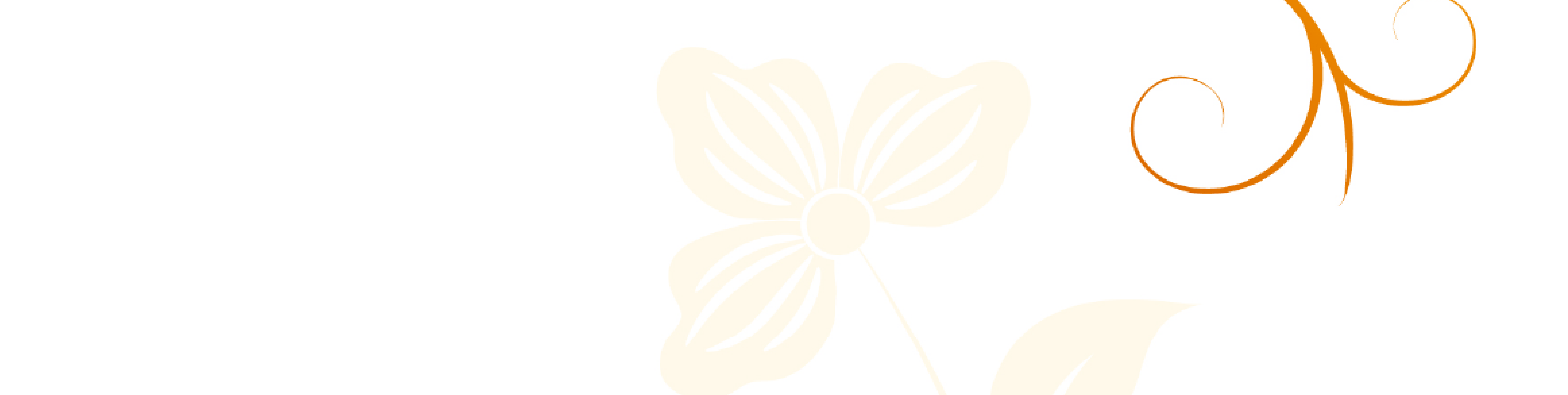
21世纪高等学校规划教材 | 计算机应用





动态网站设计与开发实用教程

李宝敏 主编
黄玉蕾 林 青 副主编

清华大学出版社

21世纪高等学校规划教材·计算机应用

动态网站设计与开发实用教程

李宝敏　主编

黄玉蕾　林　青　副主编

清华大学出版社

北　京

内 容 简 介

全书以 ASP、JSP 动态网站编程技术为主线，由浅入深、循序渐进地介绍了 HTML、JavaScript、CSS、XML 等技术，主要包括 Web 开发技术概述、HTML 语言基本结构、JavaScript 的使用、CSS 样式表、使用 ASP 和 JSP 创建动态网页、数据库的使用、网站开发方法、使用模块化方法制作主页，以及网站的建设与管理、网站应用程序实例等内容。

本书以应用为目的，以案例为引导，结合知识点案例分析，配合大量实训的演练，加强对学生实际操作与应用能力的培养，最终使学生掌握构建动态网站的知识。

这是一本适应各个专业的本、专科学生需要学习网站设计、开发和管理的公共教科书，尤其适应于那些培养应用型人才的学校，是学生学习网站知识的实用教材。本书也可以作为自学成才的读者或其他相关科技人员的参考书。

图书在版编目(CIP)数据

动态网站设计与开发实用教程/李宝敏主编. —北京：清华大学出版社，2012.1
(21 世纪高等学校规划教材・计算机应用)
ISBN 978-7-302-26959-5

Ⅰ. ①动…　Ⅱ. ①李…　Ⅲ. ①网页制作工具－程序设计－高等学校－教材
Ⅳ. ①TP393.092

中国版本图书馆 CIP 数据核字(2011)第 198303 号

责任编辑：梁　颖　王冰飞
责任校对：白　蕾
责任印制：

出版发行：清华大学出版社　　**地　址：**北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编：**100084
社　总　机：010-62770175　　**邮　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　销：全国新华书店
开　本：185×260　**印　张：**22.25　**字　数：**545 千字
版　次：2012 年 1 月第 1 版　**印　次：**2012 年 1 月第 1 次印刷
印　数：1～　000
定　价：　.00 元

产品编号：040524-01

编审委员会成员

（按地区排序）

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程”（简称“质量工程”），通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程（群）加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”（以下简称“编委会”），旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性（新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量）、先进性（对原有的学科体系有实质性的改革和发展，顺应并符合 21 世纪教学发展的规律，代表并引领课程发展的趋势和方向）、示范性（教材所体现的课程体系具有较广泛的辐射性和示范性）和一定的前瞻性。教材由个人申报或各校推荐（通过所在高校的“编委会”成员推荐），经“编委会”认真评审，最后由清华大学出版

社审定出版。

目前，针对计算机类和电子信息类相关专业成立了两个“编委会”，即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括：

（1）21世纪高等学校规划教材·计算机应用——高等学校各类专业，特别是非计算机专业的计算机应用类教材。

（2）21世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

（3）21世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

（4）21世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

（5）21世纪高等学校规划教材·信息管理与信息系统。

（6）21世纪高等学校规划教材·财经管理与应用。

（7）21世纪高等学校规划教材·电子商务。

（8）21世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力，在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌，为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格，这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会
联系人：魏江江
E-mail：weijj@tup.tsinghua.edu.cn



网站技术日新月异的发展已成为人们有目共睹的事实,而动态网站以其功能多样、维护便捷、交互性强等特点,成为大型网站(门户网站、论坛、社区等)建站的首选技术。动态网站技术涉及网站规划与管理、网页设计与制作、数据库、JavaScript、VBScript、ASP、JSP、PHP等诸多技术和方法,必要的理论讲解和大量的上机实践是学好“网站设计与开发”及其相关课程的关键。

编者针对具备“数据库技术及应用”知识基础的读者编写本书,目标是利用尽量少的课时,读者能够学习建立动态网站的相关知识并完成一定程序的实践,最终可以独立完成网站设计、开发与管理的相关任务。

本书的编写是通过知识的整合与内容的筛选,从网站设计与开发知识海洋中选其精华为我所用。以已开发的网站项目为基础,通过大量实例的演练,加强实际操作与应用能力的培养,最终使得学生掌握这门知识。对于更多、更深入的网站设计与开发的问题,学生可在实际应用中根据需要再进行知识的补充。随着网络应用的普及,各个专业都需要利用网络这个信息平台拓宽和开发本专业应用,许多在校的非计算机专业的学生经常咨询和要求开设动态网站设计与开发这门课程,因此网站(网页)的设计与开发已经成为一门计算机普及的技术,适应各个专业的本、专科学生学习网站设计、开发和管理开设的公共课。尤其对培养应用型人才的院校,学生无须花费过多的课时,不需要过深的知识铺垫,经过学习就能进行网站(网页)设计。

在本书中共设置了5部分内容,分别对动态网站的设计与开发进行了介绍。第一篇是编程基础,有3章的内容。第1章 Web 开发技术,主要介绍动态、静态网页的基本概念,以及建立动态网站相关技术,如 ASP、JSP 技术知识等。第2章介绍网站上普遍应用的超文本标记语言——HTML,通过其网页结构、语言内容、设置、实现超链接,最终达到能够独自设计并实现网站的注册页面。第3章详细介绍 JavaScript 脚本语言的特点、运行原理,以及变量、数据类型、表达式,函数等。第二篇是 ASP、JSP 基础,有5章的内容。第4章主要介绍不需要实例化过程就能直接使用的 ASP 的6个内置对象以及这些对象所提供的属性、方法和集合。第5章介绍 ASP 常用的内置组件中的广告组件、内容组件、计数器组件等使用方法。第6章介绍 JSP 指令、JSP 脚本元素和 JSP 的标准动作等 JSP 的语法规则,以及用于描述 JSP 程序代码注释和 JSP 生成的页面,即 HTML 和 XML 注释。第7章主要介绍 Servlet 技术及使用,JavaBean 技术和使用 JavaBean 标签如何进行开发等知识。第8章讲述 JSP 的5种内置对象,每一种对象所具有的方法及相应的使用说明,使用这些内置对象开发 JSP 应用程序的基本要求。第三篇是数据库开发,有3章的内容。第9章主要介绍 Access 和 SQL Server 数据库开发应用基础,以及如何应用到 Web 开发中的相关知识。第10章主要对 SQL Server 数据库与 ASP 技术相结合开发“网上在线考试系统”的案例做了详细的介绍。第11章讲述一个利用 JSP 技术完成的具有一定实用性的数据库信息管理系统——物资管

理系统的案例。第四篇是新技术与网站管理，有 3 章的内容。第 12 章讲述 XML 的发展、基本概念、XML 语法，讨论与 HTML 语言的不同之处，继而引出使用层叠样式表相关联的 XML 文件。第 13 章介绍 .NET 的核心技术——CLR 与统一的类库集，.NET 的应用开发，以及 ASP 与 ASP.NET 的区别。第 14 章针对网站的设计，介绍网站界面设计技巧、网页版式设计、运用色彩进行界面设计以及对网站的测试方法等，还讲述了域名的种类和申请方法，推广自己网站的相关技巧。第五篇是实训练习，通过 12 个实战实例的练习，对前面所讲授知识加深理解，进而达到掌握的目的。

参加本书编写的李宝敏、黄玉蕾、林青教师都是在培华学院多年从事网站(网页)设计及相关课程教学的一线骨干教师，具有丰富的教学经验。李宝敏教授负责对全书的组织与统稿。

由于本书是一次知识整合的大胆尝试，虽然我们尽力将书稿做到尽善尽美，但是限于水平，不可避免仍会存在着许多欠佳的地方，恳请同行、专家读者给予批评指正。

对于在全书的编写、出版过程中给予我们帮助、指导的教师和同仁，在此深深地表示诚挚的感谢！

编　者

2011 年 10 月



第一篇 编程基础

第三篇 数据库开发

第一篇

编程基础

第 1 章 Web 开发技术

随着 Internet 的普及，越来越多的企业建立了自己的 Web 网站。目前主流的 Web 开发技术有很多，如 ASP、ASP. NET、JSP、PHP 等，这些开发技术各有各的特点。学习这些 Web 技术，最基础的就是学习 HTML 语言，构成静态的文档，然后向 HTML 文档中添加各种动态效果语言，实现动态网站的构建。在本章的学习中，不仅要搞清楚动态网站的运行原理，还要学会配置和安装这些动态环境。

学习目标及要求

(1) 了解什么是 Web 及 Web 的工作原理。

(2) 了解什么是动态网页和静态网页技术。

(3) 了解主流的网页开发技术，重点掌握 ASP 技术和 JSP 技术。

(4) 熟悉 ASP 的运行环境，重点掌握对 IIS 的配置与安装。

(5) 熟悉 JSP 的运行环境，掌握 JDK 和 Tomcat 的安装方法。

(6) 熟练使用 Dreamweaver 软件开发动态网页。

(7) 理解什么是虚拟目录。

1.1 什么是 Web

1.1.1 Web 的起源

Internet 起源于 1969 年美国国防部高级研究计划署开发的 ARPANET 网。1985 年在美国国家科学基金会的推动下，将军事用途转向科学研究，最终形成了今天网络的雏形。从 1994 年开始，我国计算机网络中心正式接入 Internet，并且得到了迅速发展，网民无数。

网络目前提供的服务有 WWW、E-mail、FTP、Telnet 等。

WWW(world wide web)又称万维网，1989 年由欧洲物理研究室开发，最初意愿是为了方便研究人员相互传递资料。自从 WWW 开发成功之后，Internet 迅速进入了千家万户，成为人们学习、工作、交流、娱乐的一个非常重要的手段。最初的 WWW 网页主要用来呈现一些静态信息，如信息介绍、新闻、图片等，采用超文本标记语言(hypertext markup language，HTML)来实现(HTML 将在第 2 章介绍)。人们可以通过各种 HTML 标记实现文本、图像、超链接、表格等内容。

尽管 HTML 非常简单实用，但不方便更新，比如，修改网页内容，只能修改 HTML 文

件，如果遇到文字和内容很多的时候，不仅不利于修改，而且难于维护。人们就考虑，能否有一个更好的方式解决这样的问题，于是动态网络程序设计语言就应运而生了。

1.1.2 Web 的客户-服务器模型

WWW 的工作原理就是客户-服务器模型，如图 1-1 所示。通常将提供服务的地方叫做服务器端，接受服务的地方叫做客户端。

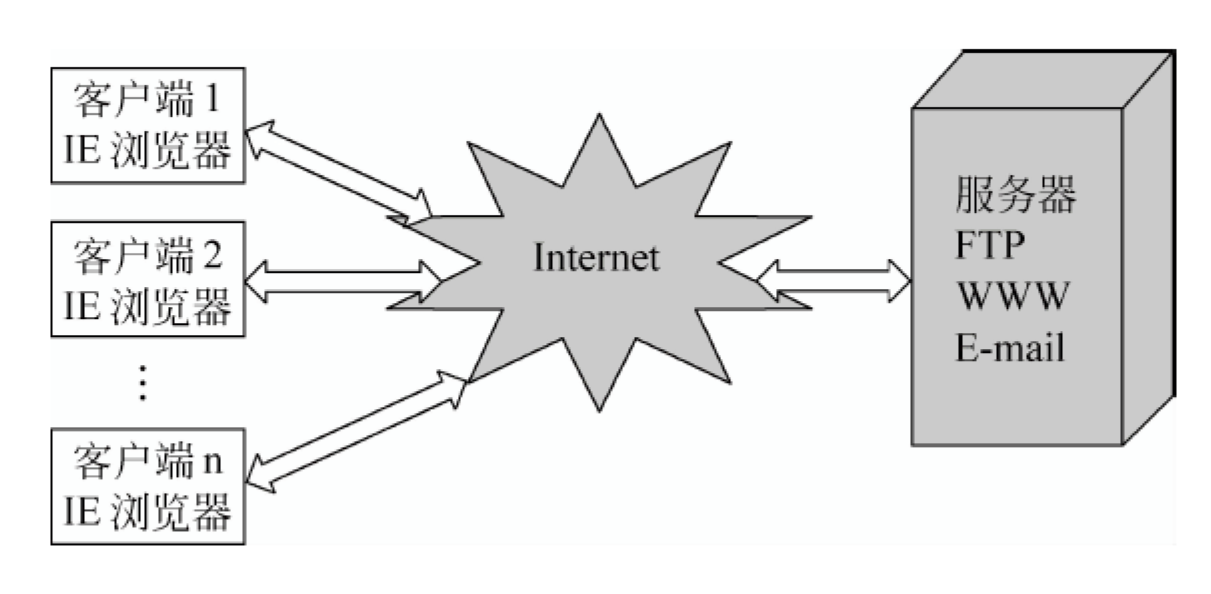


图 1-1 WWW 的工作原理

通过浏览器可以访问新浪、中华网、淘宝网、网易、搜狐以及微软公司的网站等，这些就是 Web 应用。对于 Web 应用，需要使用浏览器，通过网络访问在远程的服务器上运行的程序。

上网的一般过程如下：打开浏览器，输入某个网址，经过一段时间的等待，浏览器显示要访问的信息，然后可以在网页上继续进行其他操作。在网页上点击超链接，访问希望访问的内容，等待浏览器中内容的再次更新。实际上，这些都是客户端和服务器端进行交互访问。

1.1.3 简单静态网页和动态网页

静态网页是指由 HTML 标记语言组成的，后缀为.html 或.htm。静态网页只能显示内容，不能动态修改内容。如果需要修改，必须修改.HTML 或.HTM 源文件，再重新上传到服务器。

静态网页的工作原理如图 1-2 所示。

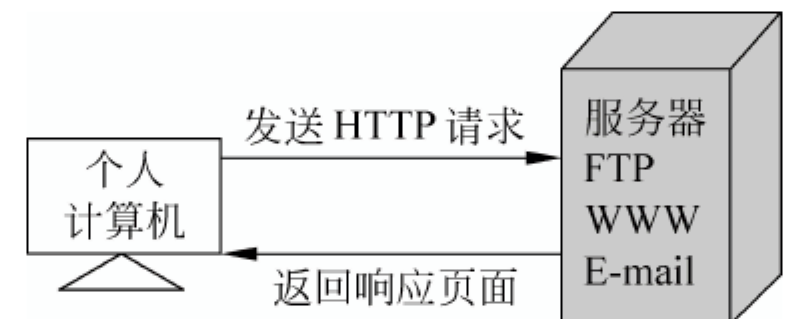


图 1-2 静态网页工作原理

个人用户在计算机中使用 IE 浏览器，向服务器发送 HTTP 请求，请求访问某一页面，然后服务器接受请求，并找到静态页面，发送给个人计算机。

动态网页主要由程序设计语言组成，能够动态地修改网页内容，比如留言板、网络聊天室、新闻发布等。这种程序设计语言有 Java、C＃、VB、C++ 等。本书主要介绍以 ASP 和 JSP 两大技术为主的动态网页开发技术。

动态网页的工作原理：用户在浏览器的地址栏中输入 ASP 文件，并回车触发这个 ASP 的申请。浏览器将这个 ASP 的请求发送给 Web 服务器。Web Server 接收这些申请并根据.asp 的后缀名判断确定是 ASP 请求。Web Server 从硬盘或内存中读取对应的 ASP 文件，并将这个文件发送到 ASP.dll 文件中处理。ASP 文件将会从头至尾执行并根据命令要

求生成相应的 HTML 文件，新编译 HTML 文件被送回浏览器，用户的浏览器解释这些 HTML 文件并将结果显示出来。

1.2 流行的 Web 开发技术

上网的时候，在地址栏中可以看到文件的后缀名，可根据后缀名知道网站采用什么技术。比如，当浏览清华大学招生网站的时候，看到地址栏中文件名的后缀是.jsp。如图 1-3 所示的界面是其中的一个页面，地址是 http://join-tsinghua.edu.cn/jsp/lqqk/lqcx.jsp，这个地址说明网站使用的是 JSP 技术。



图 1-3 JSP 的页面

京东网站是一个大型的电子商务网站，网页的后缀名是.aspx。如图 1-4 所示的界面是京东网站的订单页面，地址是 http://jd2008.360buy.com/JdHome/OrderList.aspx，这说明网站使用的是 ASP.NET 技术。

1.2.1 ASP 技术

ASP(active server pages，活动服务器网页)是服务器端脚本编写环境，可以创建和运行动态、交互、高效的 Web 服务器应用程序。使用 ASP 可以组合 HTML 标记、脚本命令和 ActiveX 组件，以创建交互的 Web 页和基于 Web 的功能强大的应用程序。

ASP 文件是以.asp 为扩展名的文本文件。

在 ASP 文件中通常包含文本、HTML 标记和脚本命令，这三部分内容以各种组合混杂



图 1-4 ASP.NET 的页面

在 ASP 文件中，需要使用不同的符号进行区分。

注意：

(1) 在 ASP 程序中，字母不区分大小写。

(2) 在 ASP 中，凡是在语法中用到标点符号的，都是在英文状态下输入的标点符号，否则将出错。只有一种情况除外，就是在字符串中使用的标点符号。

(3) 一条 ASP 语句就是一行。

(4) 要养成良好的书写习惯，比如恰当的缩进。

(5) 在 ASP 程序中可以包含 HTML 语言，两者相互配合使用。

1.2.2 JSP 技术

JSP(Java server pages)是 Sun 公司于 1999 年推出的网站开发技术，有多个公司参与和支持，包括 IBM、Oracle、BEA 等公司，还有一些社团，包括著名的 Apache 组织。JSP 主要采用 Java 技术完成。JSP 在 HTML 文件中加入 Java 程序片段和 JSP 标记，构成了 JSP 网页。在技术方面，JSP 与 ASP 非常相似。

JSP 最大的优点是开放的、跨平台结构，可以运行在几乎所有的服务器系统上，包括 UNIX、Linux、Windows 等。

JSP 使用的是 Java 语言，它继承了 Java 语言的特点——“一次编写，处处运行”，这种概念正越来越深远地影响着因特网行业的交互式 Web 页面的设计理念。JSP 页面可以非常容易地跨平台、跨 Web 服务器软件来设计和开发源代码。

JSP 组件(企业 JavaBeans、JavaBeans 或定制的 JSP 标签)都是跨平台可重用的。企业 JavaBeans 组件可以访问传统的数据库，并能以分布式系统模式工作于 Solaris、Linux、UNIX 和 Windows 平台。

目前，JSP 技术支持的网页格式还没有一个明确的标准。一般来说，JSP 技术既可以支持 HTML/DHTML 的传统浏览器文件格式，又可以支持应用于无线通信设备如移动电话、PDA 等设备进行网页预览的 WML 文件格式，还可以支持其他一些 B2B 电子商务网站应用的 XML 格式。

尽管ASP和JSP都使用标签与脚本技术来制作动态Web网页,但JSP技术允许开发者扩展JSP标签、定制JSP标签库,所以网页制作者充分利用与XML兼容的标签技术强大的功能,大大减少对脚本语言的依赖。

总的来说,JSP和ASP在技术方面有许多相似的地方。它们都是基于Web的应用程序开发方案,它们都是在Web服务器端执行的程序。ASP在网页制作中占据了一定的地位,但是其本身还有很多的缺点。JSP出现在ASP之后,兼顾了ASP的很多优点,同时也克服了ASP的一些缺点,是一种比较成熟的网页开发程序语言。

1.2.3 ASP.NET技术

ASP.NET是创建动态网页的一种强大的服务器端新技术。它可为WWW站点或为企业内部互联网创建动态的可进行交互的HTML页面。它采用面向对象的方法来构建动态Web应用程序。ASP.NET可以用来建设门户网站,实现复杂的基于Web的系统、电子邮件发送系统等。可在Internet或Intranet上部署ASP.NET Web应用程序。

1.2.4 脚本语言JavaScript和VBScript

Netscape公司为了进一步扩充它的浏览器的功能,开发了一种可以嵌入在Web主页中的编程语言。在早期这种语言叫做LiveScript,后来为了利用Sun公司开发的Java语言的功能并借用它的流行性,把它改名为JavaScript。

为了运用JavaScript控制HTML页面上的对象,JavaScript的代码必须与HTML代码结合在一起。将JavaScript嵌入HTML页面时,必须使用<SCRIPT>标签,

VBScript是Visual Basic Script的简称,即Visual Basic描述语言,有时也被缩写为VBS。VBScript是微软开发的一种脚本语言,广泛应用于网页和ASP程序制作中,同时还可以直接作为一个可执行程序,用于调试简单的VB语句也非常方便。

VBS是微软的Active Server Pages的一部分,它与Java Server Pages和PHP是竞争对手。在这里,VBS的程序码直接嵌入到HTML页内,这样的网页以ASP结尾。Internet网页服务器(IIS)执行ASP程序,并将其结果转化为HTML传递给网页浏览器,以供用户使用。

VBScript的主要优点是:VBScript的操作系统不是由网页浏览器解释,它的文件比较小、易学,在所有2000/98 SE以后的Windows版本都可直接使用。

VBScript的缺点是:现在VBS无法作为电子邮件的附件了。由于Microsoft Outlook拒绝接受VBS为附件,所以收信人无法直接使用VBS附件。

1.2.5 PHP技术

PHP是Rasmus Lerdorf在1994年提出来的、用PERL语言编写的程序,主要用来和访问其他主页的人保持联系。

PHP将脚本嵌入到HTML文档中,采用了C、Java、PERL语言的语法,加入了自己的特征,在服务器端运行。

PHP是免费的开放源代码软件，可以进行免费复制、编译，但PHP安装运行环境复杂，不利于学习。

1.3 ASP的运行环境

服务器端需要安装如下软件：

(1) Windows 98或Windows 2000 Professional或Windows 2000 Server或Windows 2000 Advance Server或Windows XP Professional或更高版本，其中，Windows 2000系列需要安装Service Pack 2.0。

(2) IIS 5.0(Internet信息服务管理器5.0)或更高版本。

客户端只要是普通的浏览器即可，如Internet Explorer 5.0或更高版本。

注意：对于初学者而言，只须将自己的计算机同时作为客户端和服务器端，无须搭建相应的网络环境。

1.3.1 安装IIS

安装方法：

依次选择“开始”→“设置”→“控制面板”→“添加/删除程序”命令，在“添加或删除程序”对话框中选择“添加/删除Windows组件”按钮，就会弹出“Windows组件向导”对话框，如图1-5所示。在其中选择“Internet信息服务(IIS)”，然后单击“下一步”按钮，随后根据提示一步步安装即可，如图1-6所示。



图1-5 IIS的安装——Windows组件向导



图 1-6 IIS 的安装

安装完毕后，选择“开始”→“设置”→“控制面板”→“管理工具”→“Internet 信息服务”，打开“Internet 信息服务”窗口，进行预览，IIS 服务安装完毕，如图 1-7 所示。



图 1-7 管理工具

1.3.2 IIS 默认网站

安装完毕后，在 IE 浏览器中输入“http://localhost”，如果能显示 IIS 欢迎字样，就表示 IIS 安装成功。依次选择“开始”→“设置”→“控制面板”→“管理工具”→“Internet 信息服务”命令，就会出现“Internet 信息服务”的窗口，如图 1-8 所示。IIS 安装完毕后，会自动生成 C:\inetPub\wwwroot，以后制作的网页文件存放在该文件夹下，就可以访问了。



图 1-8 默认网站的文件显示

1.3.3 IIS 的启动和停止

启动 IIS 时，直接进入“Internet 信息服务”，右击个人的首页文件，例如，index.asp 文件，从弹出的快捷菜单中选择“浏览”命令，就可以运行 ASP 程序了，如图 1-9 所示。

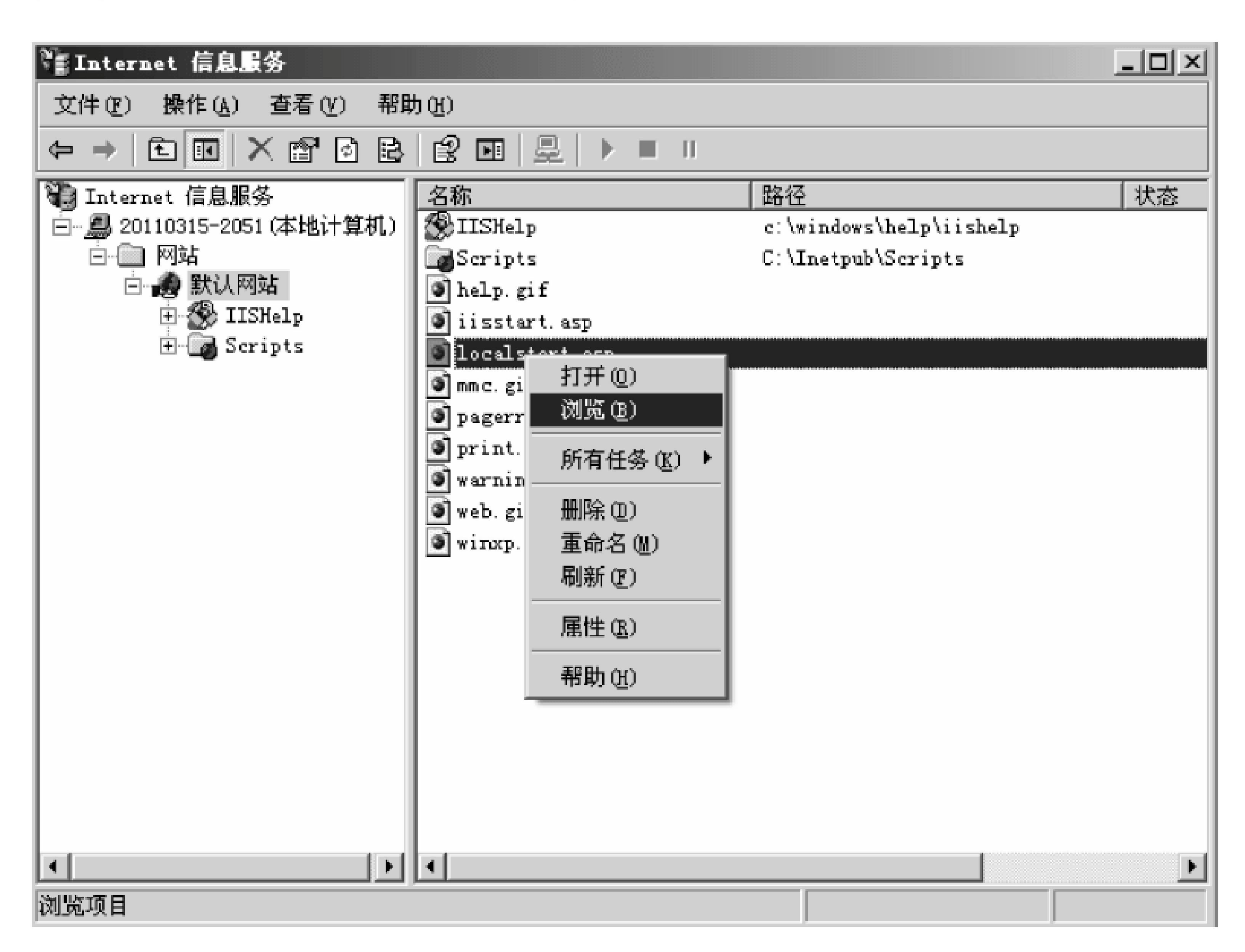


图 1-9 浏览的操作方法

打开“Internet 信息服务”窗口，在窗口左侧的树型窗格中的本地计算机图标上右击，从弹出的快捷菜单中选择“所有任务”→“重新启动 IIS”命令。

停止时，需要关闭 IIS 服务。

1.3.4 IIS属性的设置

在网站使用中，需要对IIS的常用属性进行设置分配。比如网站的IP地址分配、TCP端口等。一般情况下，网站端口很容易和一些杀毒软件的端口产生冲突，所以需要修改此处的端口数字，更改成其他数字即可，如图1-10所示。

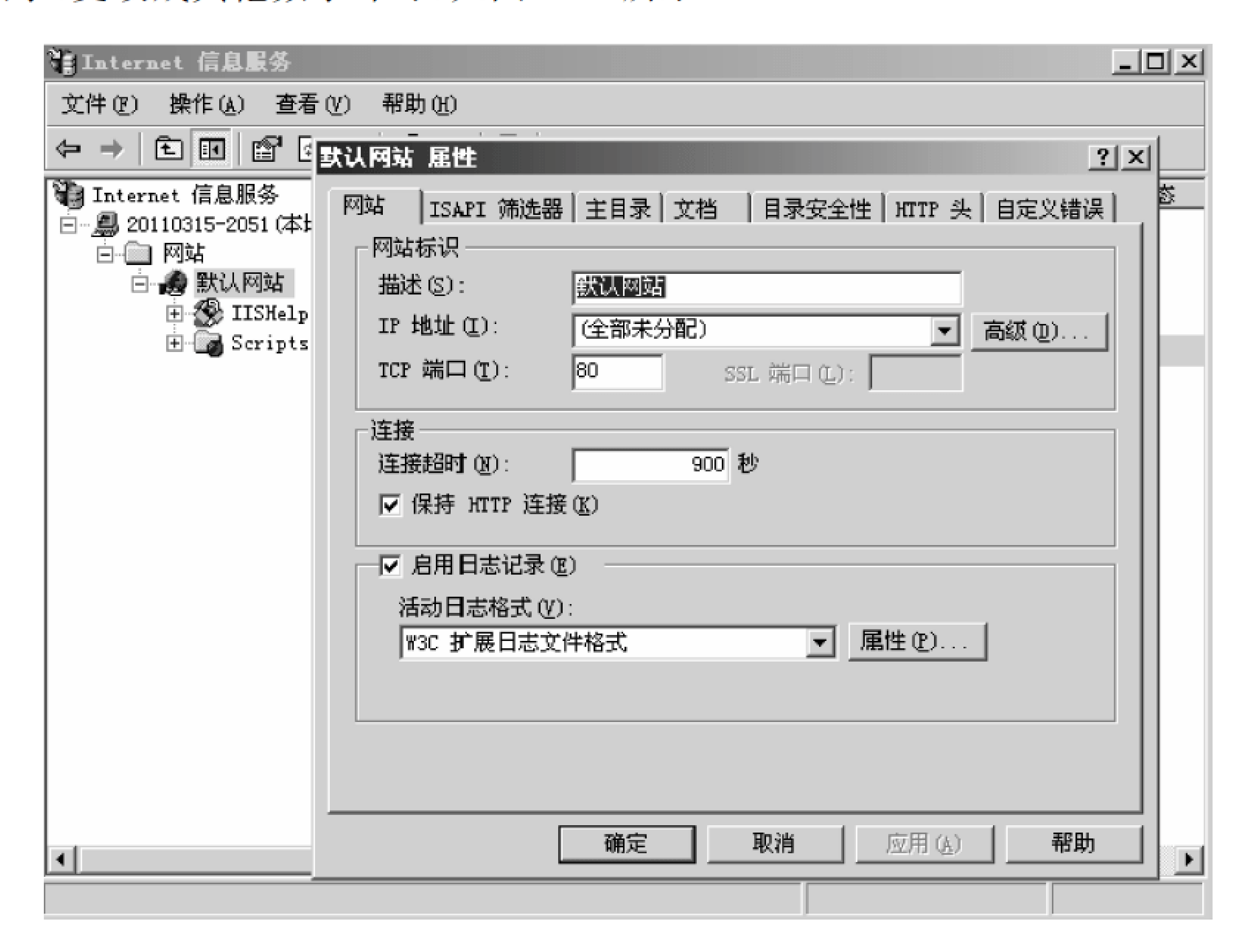


图1-10 设置网站标识

在IIS属性中，有时需要设置主目录信息。当网站程序不可显示时，有一种情况可能是目录路径地址错误，需要重新定位地址。方法是选择本地路径，单击“浏览”按钮，找到自己的源文件路径地址，然后单击“确定”按钮，如图1-11所示。

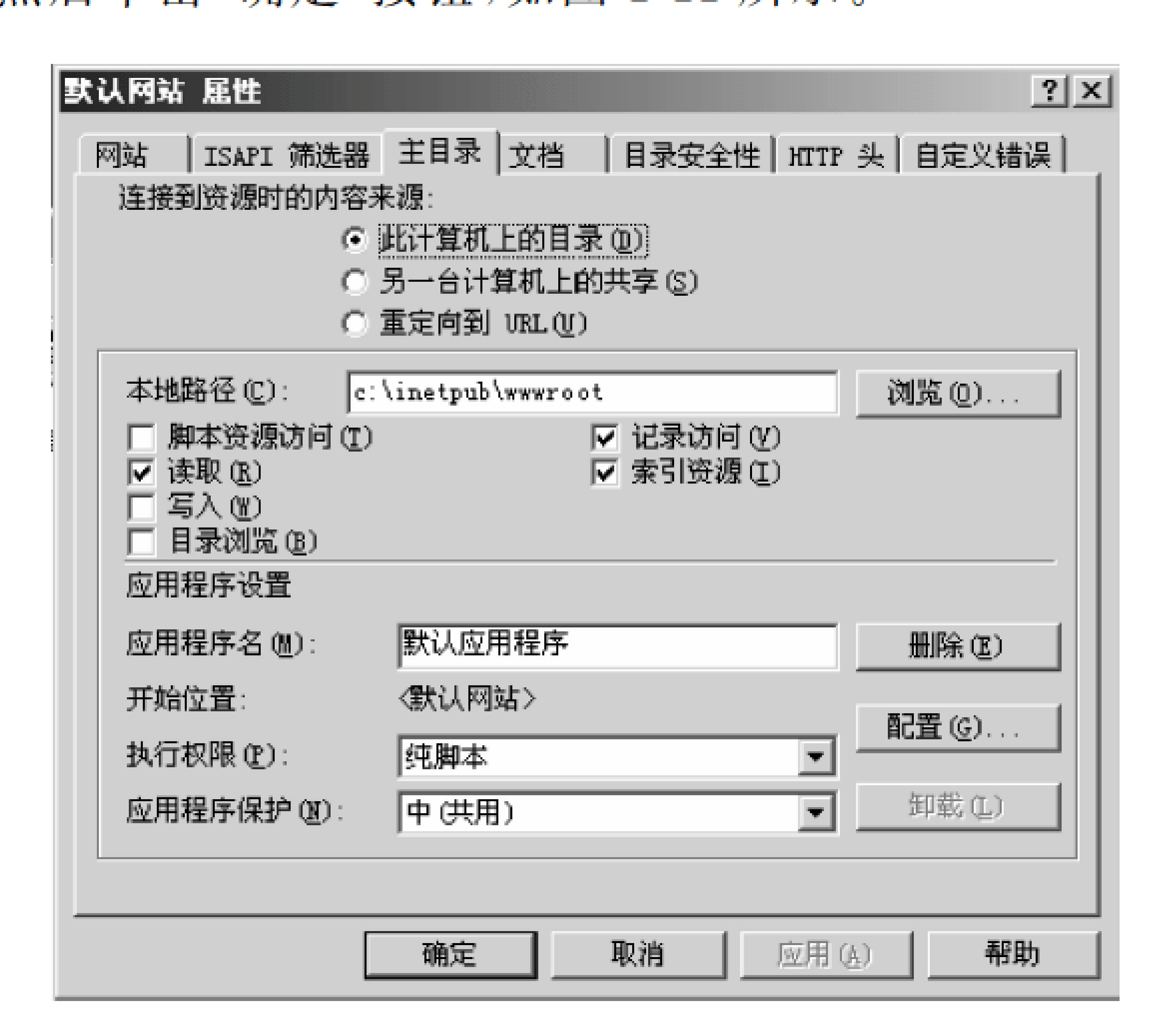


图1-11 设置网站主目录

在文档属性中列举了当前网站的 ASP 页面名，也可自己添加，主要作用是决定网站的浏览页的先后顺序，如图 1-12 所示。



图 1-12　设置网站文档

1.3.5　IIS 虚拟目录的创建

虚拟目录是指物理上没有包含网站文件，但 IIS 中却将该网站文件设置成主目录下的文件。特点是网站文件虽然在 D、E、F 等盘符下，但浏览器在浏览时，并不是直接访问这些盘符，而是访问 IIS 的主目录。访问时，只须输入 localhost＋文件名，这样大大减小了文件地址的长度，使得访问更加简单方便。比如 http://localhost\onlineexam\index.asp。

下面给出了使用 IIS 管理器创建虚拟目录的方法，如图 1-13 所示。

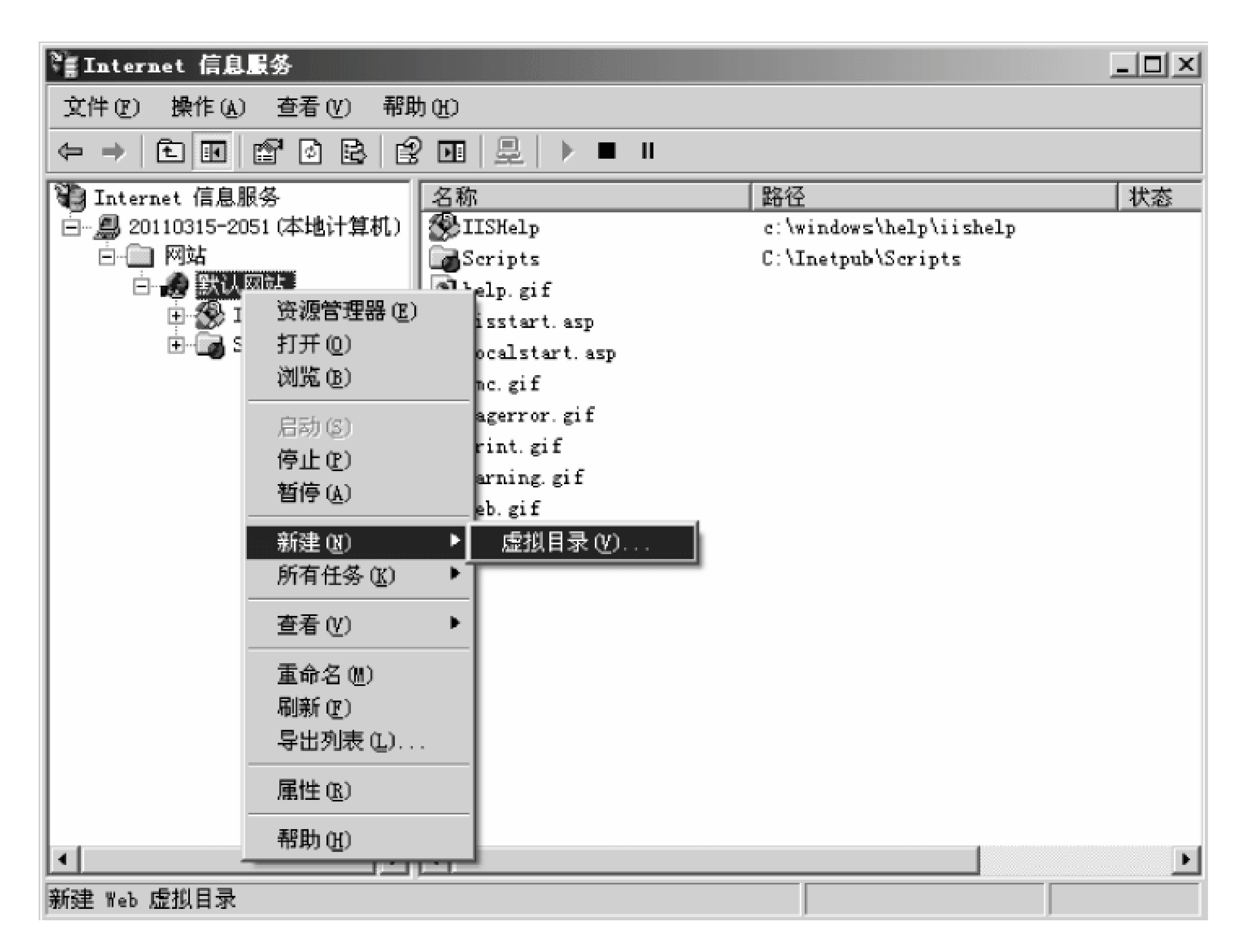


图 1-13　虚拟目录的创建

虚拟目录的创建向导如图 1-14 所示。设置虚拟目录的别名如图 1-15 所示。设置网站对应的物理地址如图 1-16 所示。设置虚拟目录的访问权限如图 1-17 所示。提示虚拟目录安装完成如图 1-18 所示。

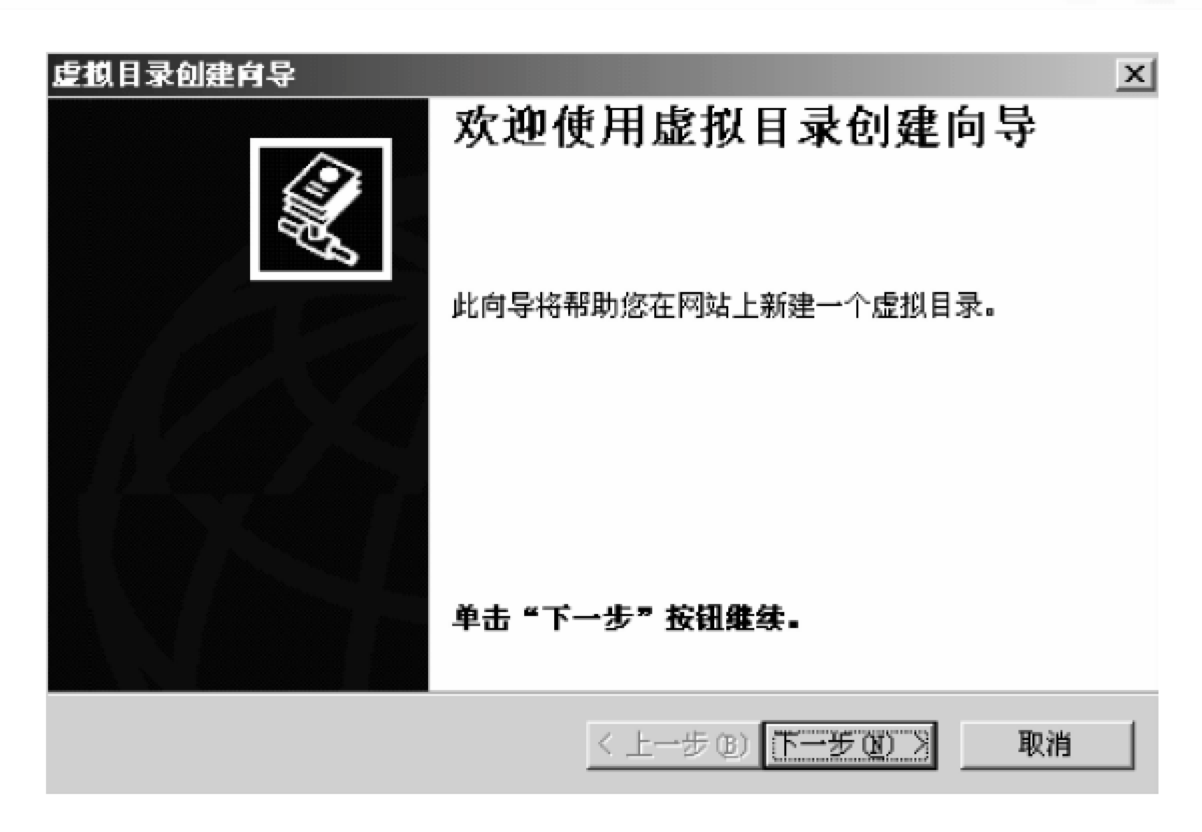


图 1-14　虚拟目录创建向导

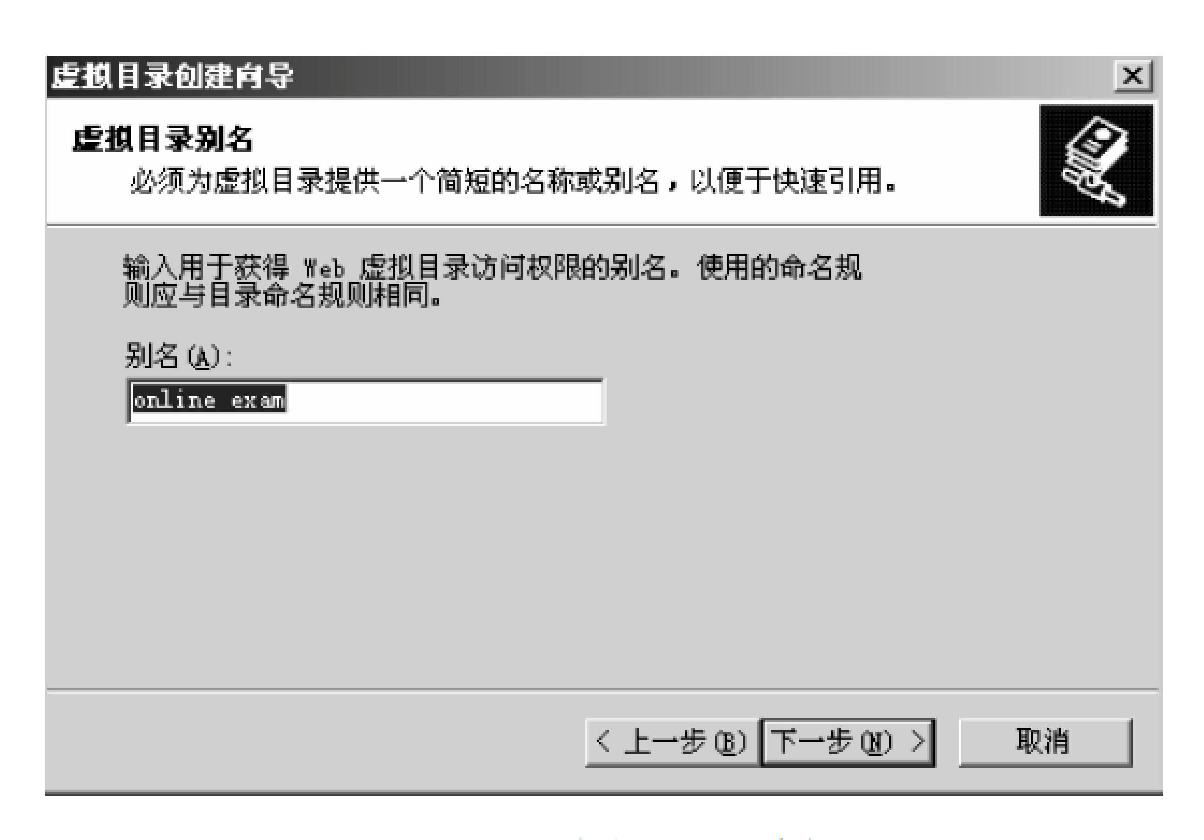


图 1-15　虚拟目录别名



图 1-16　网站对应的物理地址

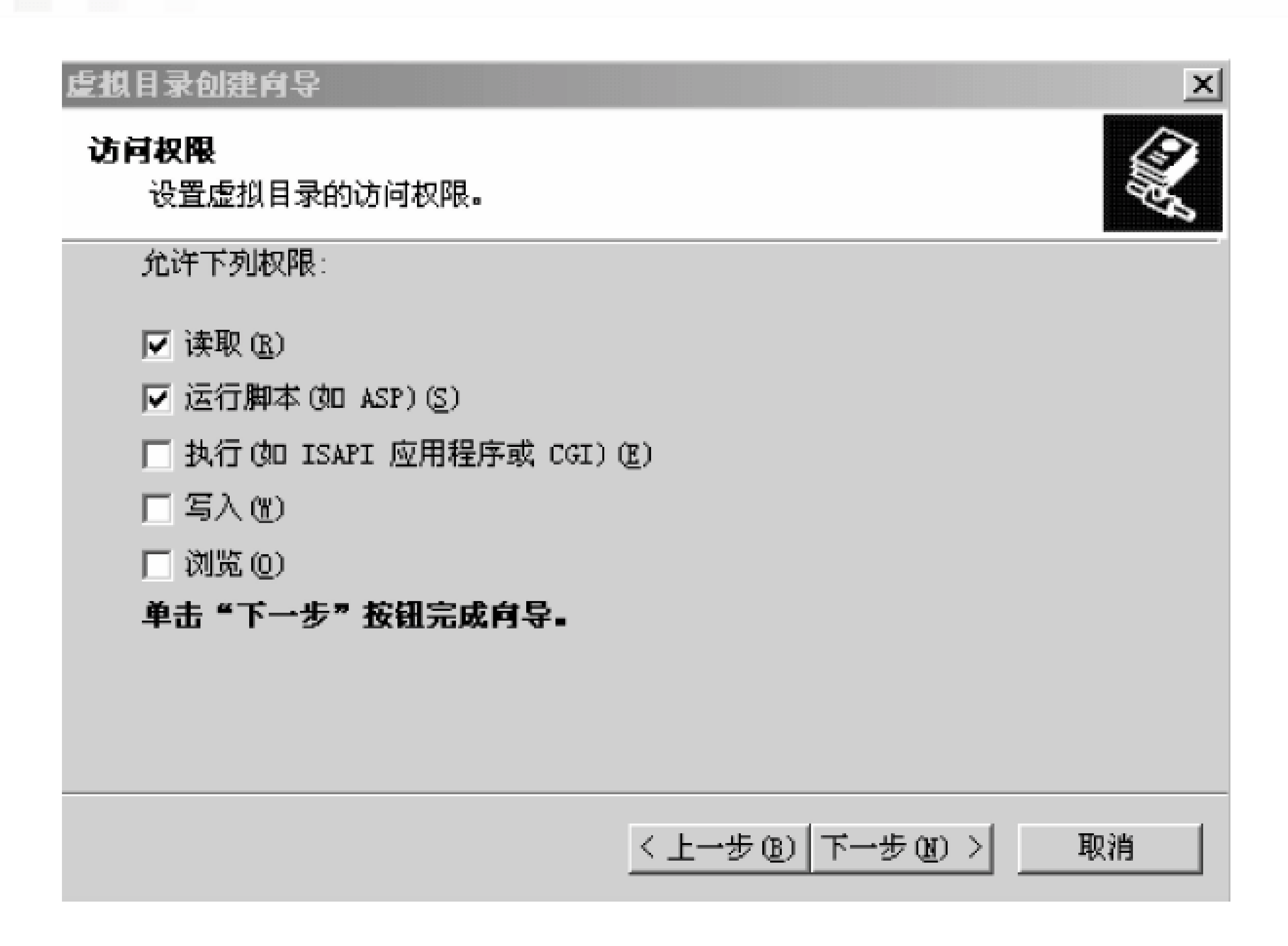


图 1-17 访问权限



图 1-18 安装完成

项目案例 1:IIS 环境的配置方法。

具体操作步骤:

(1) 首先在 E:\下创建一个文件夹,名字叫 Web。然后在该文件夹下创建两个文件夹,一个叫 images,另一个叫 css。Web 文件夹下存放 asp 文件。

(2) 依次选择“开始”→“控制面板”→“管理工具”→“Internet 信息服务”命令,打开“Internet 信息服务”对话框。右击其中的“默认网站”,从弹出的快捷菜单中选择“新建”→“虚拟目录”命令,然后按提示操作,添加别名“my Web”,并为其选择对应的文件夹“E:\Web”,最后单击“完成”按钮即可。

(3) 右击新添加的虚拟目录“my Web”,从弹出的快捷菜单中选择“属性”命令,就会出现“属性”对话框。在其中添加 index.asp、default.asp 等默认文档并调整顺序后,单击“确定”按钮即可。

(4) 现在就可以在 Web 文件夹下建立 ASP 文件了,如建立一个计算两个数的和的文件,保存在 Web 文件夹下,名称为 index.asp。



1.4　JSP的运行环境

运行JSP需要安装JDK和一个JSP服务器，目前比较流行的JSP服务器是Tomcat、Weblogic、WebSphere和Resin。JSP运行对CPU和内存要求比较高，尤其是开发的过程。目前常规的CPU已经完全可以满足需求，P4以上的CPU就可以完全胜任开发和运行工作了。对内存的要求建议使用1GB以上的内存，尤其是开发数据库的项目需要大内存，不然调试程序时的效率会很低。

操作系统环境可以选择Windows操作系统，也可以采用UNIX家族的桌面系统。推荐使用Windows XP/ 2003 Server操作系统。尽量和以后系统发布的环境一致，这样就可以尽量避免出错的可能性。其实Linux和FreeBSD下的KDE环境对Eclipse支持得也很好，比如在FreeBSD6.1＋KDE3.5＋Eclipse3.2平台下开发。

1.4.1　下载与安装JDK

JSP运行环境需要JDK的支持，要安装两个软件：j2sdk和Tomcat。安装了JDK以后，就建立好了J2SE的运行环境，这里安装JDK5，也叫JDK1.5。JDK的显示界面如图1-19所示。



图1-19　JDK的显示界面

1.4.2　下载与安装Tomcat

在默认的情况下，Tomcat安装在系统盘的Program Files目录下，比如“Program Files\Apache Software Foundation\Tomcat 5.5”，安装完毕的目录结构如图1-20所示。

1.4.3　运行JSP程序

第一个程序依然是最经典的“Hello World!”程序，在记事本中输入例1-1中的代码，如下所示。JSP区分大小写，每条语句后面有分号，将该程序放到D:/java/ROOT目录下，然后输入“http://locahost:8090/2-01.jsp”地址，显示的界面如图1-21所示。

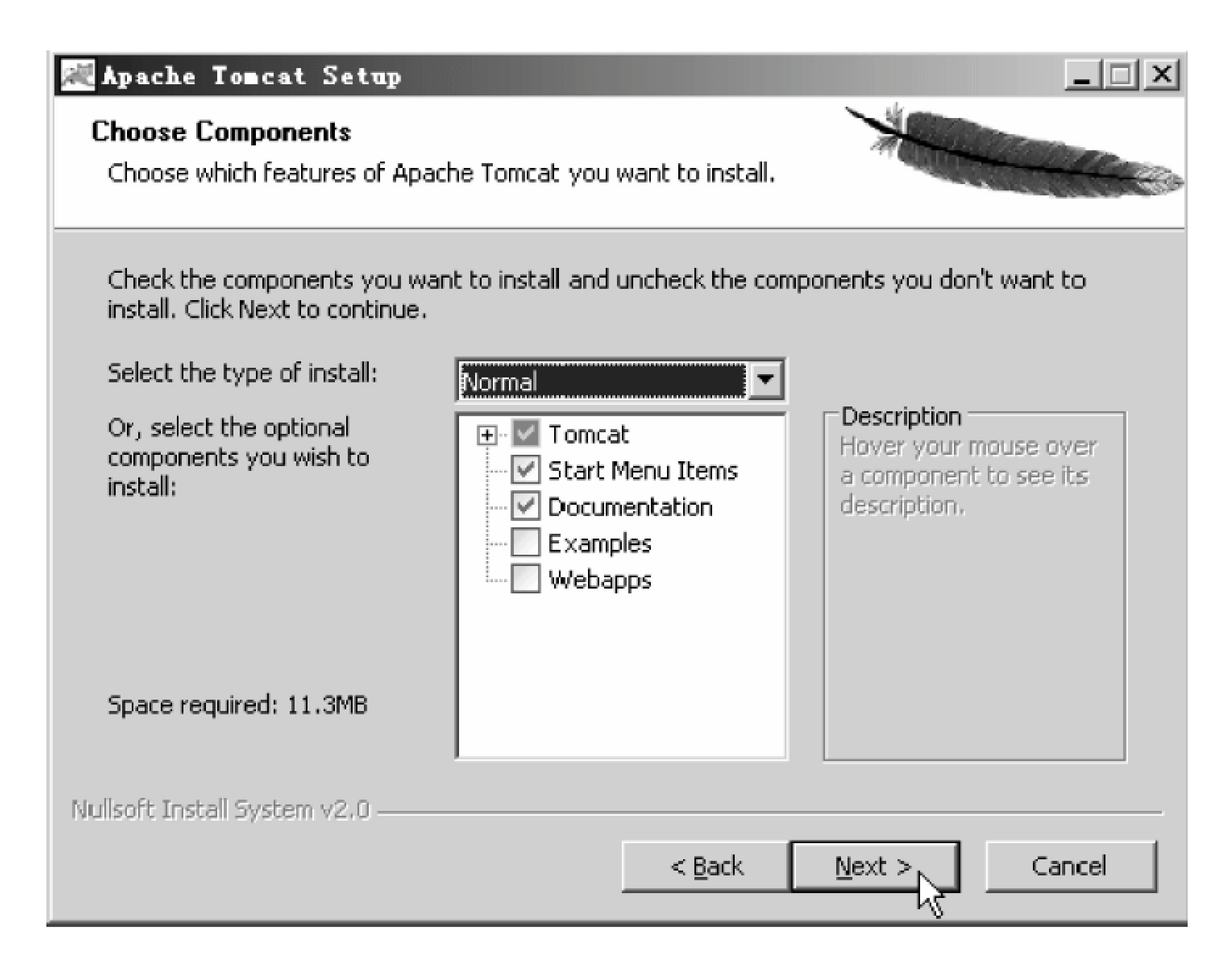


图 1-20 Tomcat 的安装



图 1-21 JSP 的输出程序

【例 1-1】 第一个 JSP 程序“Hello World!”。

```
<%
out.print("Hello World!");
%>
```

1.5 ASP 开发工具

开发 ASP 的工具很多，比较专业一点的开发工具有 Visual InterDev、FrontPage、Dreamweaver、Edit Plus 等。如果要开发大型的网站，应该使用 Visual InterDev 开发工具包。如果只是编辑简单的网页，使用 FrontPage、Dreamweaver 即可。

Dreamweaver 是建立 Web 站点和应用程序的专业工具。它将可视布局工具、应用程序开发功能和代码编辑组合为一个功能强大的工具，使每个级别的开发人员和设计人员都可

利用它快速创建界面。它是在一个集成和高效的环境中为专业人员提供的工具。开发人员可以将其选择的服务器技术与 Dreamweaver 配合使用,建立将用户连接到数据库、Web 服务和旧式系统的强大 Internet 应用程序。

1.6 本章小结

本章主要介绍 Web 的发展过程,介绍了主流的 Web 开发技术。要求大家重点学习 ASP 的 IIS 安装,以及 JSP 的 Tomcat 安装方法。

选择哪一种 Web 应用程序技术取决于多个因素,其中包括对各种脚本语言的熟悉程度以及要使用的应用程序服务器。

- 如果采用 ColdFusion MX 服务器,则可以选择 ColdFusion。
- 如果采用 IIS 服务器,则可以选择 ASP 或 ASP.NET。
- 如果采用 PHP 服务器,则可以选择 PHP。
- 如果采用 JSP 服务器,则可以选择 JSP。

在 IIS 配置中,一般按照以下步骤进行操作。

(1) 添加 IIS 组件。

(2) 打开 IIS 窗口,进入配置界面。

(3) 找到"Web 站点"选项,右击鼠标进入属性。

(4) 在属性中选择网站标签,绑定 IP 地址。

(5) 在属性中选择主目录标签,确定文件的主目录。

(6) 在属性中选择文档标签,确定网站的首页文件。

1.7 思考与练习

1. 什么是 Web 编程? Web 编程有什么特点?
2. 静态网页和动态网页有什么区别?
3. ASP 程序包括哪些内容?
4. 如何配置 IIS、发布网站?
5. 如何配置 JDK? 如何安装 Tomcat?

第2章 HTML

HTML 是一种用来制作超文本文档的简单标记语言。超文本传输协议规定了浏览器在运行 HTML 文档时所遵循的规则和进行的操作。用 HTML 编写的超文本文档称为 HTML 文档,它能独立于各种操作系统平台。使用 HTML 语言描述的文件需要通过 Web 浏览器显示出效果。本章主要介绍 HTML 页面结构,包括 HTML 标记的内容、设置文档、实现超链接等内容。

学习目标及要求

(1) 熟悉 HTML 文档的基本结构和语言。

(2) 熟练掌握文本和列表格式的使用方法,能够插入图像、表格,创建链接。

(3) 能够熟练设计 HTML 表单。

(4) 能够独自设计并实现网站的注册页面。

2.1 HTML 简介

HTML 即超文本标记语言,是使用特殊标记来描述网页文档结构和表现形式的一种语言。

HTML 最早是由 Tim Berners-Lee 在 1990 年设计,之后开始迅速流传。1997 年,万维网联盟(World Wide Web Consortium)制定了新的 HTML 3.2 标准,使得 HTML 文档在不同的浏览器和操作平台上能正确地显示。目前,HTML 已经升级到 4.0 版本。

HTML 从严格意义上讲并不是一种程序设计语言,它是由标记和属性组成的规则,而且 HTML 是纯文本的文件格式,可以用任何文本编辑器(例如记事本、EmEditor、EditPlus 等)或 Web 开发工具(例如 FrontPage、Dreamweaver 等)进行编辑。文件中的文字、字体颜色、字体大小、段落布局、图片大小、表格样式及超级链接,都可以用不同意义的标记(Tags)来描述。

注意:标记名称不区分大小写。

2.2 HTML 页面结构标记码

在 HTML 中,所有的标记都必须用尖括号(即小于号<和大于号>)括起来。HTML 文件在编写完毕后保存成扩展名为.htm 或.html 的文件。本教材中,主要以 Dreamweaver 为

主要编辑软件进行讲解。

在动手设计网页前，就好比盖大楼一样，必须把框架搭建起来，然后在里面添砖加瓦。HTML 就是一份整个网页的框架结构，只有搭建好了，才能进行设计工作。

【例 2-1】 编写一个简单的“HELLO WORLD!”的 HTML 文件。



```
<HTML>
<HEAD>
<TITLE> MY FIRST WEB! </TITLE>
</HEAD>
<BODY>
    HELLO WORLD!
</BODY>
</HTML>
```

从程序清单中可以看出，HTML 文档标记<HTML>和</HTML>、头标记<HEAD>和</HEAD>、标题标记<TITLE>和</TITLE>、文件体标记<BODY>和</BODY>都是成对出现的，也就是我们通常说的双标记。但在 HTML 中也有些标记是单标记，如换行标记
、水平线标记<HR>等。图 2-1 就是“HELLO WORLD!”文档显示效果。



图 2-1 “HELLO WORLD!”文档显示效果

一个基本的 HTML 文档通常由文档标记、头部标记和主体标记三部分组成，下面详细介绍这些标记。

2.2.1 文档标记<HTML>…</HTML>

文档标记用来标识 HTML 文件，它包括整个 HTML 文档。

文档标记的格式为：

```
<HTML> HTML 文档的内容 </HTML>
```

<HTML>处于文档的最前面，表示 HTML 文档的开始，即浏览器从<HTML>开始解释直到遇到</HTML>为止。每个 HTML 文件均以<HTML>开始，以</HTML>结束。

文档标记包含的标记有<HEAD>、<BODY>等。

2.2.2 文件体标记<BODY>…</BODY>

文件体标记是网页的主体标记，用于编写网页文件的主体。在 BODY 部分可以设置背景颜色、背景图片和主体部分的字体大小等信息。BODY 的属性如表 2-1 所示。

表 2-1 BODY 的属性

属性	功能
background=URL	设置网页的背景图片
bgcolor=colorvalue	设置网页的背景颜色
text=colorvalue	设置文本的颜色
link=colorvalue	设置尚未被访问过的超文本链接的颜色，默认为蓝色
vlink=colorvalue	设置已被访问过的超文本链接的颜色，默认为紫色
alink=colorvalue	设置超文本链接在被单击的瞬间的颜色，默认为红色
bgproperties =fixed	设置背景是否随滚动条滚动
leftmargin=size	设置网页左边的空白
topmargin=size	设置网页上方的空白
marginwidth=size	设置网页空白的宽度
marginheight =size	设置网页空白的高度

文件体标记的格式为：

```
<BODY> 主体的内容 </BODY>
```

2.2.3 文件头部标记<HEAD>…</HEAD>

文件头部包含在<HEAD>和</HEAD>之间所有的内容，尽管不显示在页面中，但是非常重要，它会告诉浏览器如何处理文档主体内的内容。

文件头部标记的格式为：

```
<HEAD> 头部的内容 </HEAD>
```

其内容可以是标题名、文本文件地址、创作信息等网页信息说明。

文件头部标记包含的标记有<TITLE>、<BGSOUND>、<META >、<SCRIPT>、<STYLE>。其中，TITLE 为标题标记，BGSOUND 为播放背景音乐，META 为描述网页信息，SCRIPT 为脚本，STYLE 为样式。

2.2.4 文件标题标记<TITLE>…</TITLE>

文件标题标记的格式为：

```
<TITLE> 标题名 </TITLE>
```

TITLE 是设定 HTML 文档标题的标记，写在头部标记之中。在文档头部定义的标题内容不在浏览器窗口中显示，而是在浏览器的标题栏中显示。

【例 2-2】 编写一个带有背景音乐的网页文件。

```
<HTML>
    <HEAD>
    <TITLE>宝贝音乐网</TITLE>
    <bgsound src="d:\music.mp3" loop="-1">
    <meta name="GENER">
    </HEAD>
</HTML>
```

项目案例 1：HTML 标记案例。

【题目】 制作在线考试网站页面，显示在线考试标题，能够播放提示考试时间的播放音乐，搭建简单框架，如图 2-2 所示。

【代码】

```
<HTML>
    <HEAD>
    <TITLE>在线考试网</TITLE>
    <bgsound src="d:\ks.mp3" loop="-1">
    <meta name="GENER">
    </HEAD>
    <BODY>
        欢迎登录在线考试网!
    </BODY>
</HTML>
```



图 2-2 在线考试网文档显示效果

注意：背景音乐的播放路径要一致。

2.3 定义文档格式的标记码

在大多数网页中，文档是核心的内容，只有设置适当的文档格式，才能得到绚丽多彩的网页。设置文档的格式包括设置标题和文字的字体、字号、字型、颜色，段落格式，文本布局等。

2.3.1 设置字号属性

HTML 文件中，不同的子标题通过<H1>～<H6>标记进行设定，以区分不同标题大小。如<H1>…</H1>表示最大的字体，<H6>…</H6>表示最小的字体，中间 H2 到 H5 依次从大到小排列。

标题标记的格式为：

```
<H1>…</H1>、<H2>…</H2>、<H3>…</H3>、<H4>…</H4>、<H5>…</H5>、<H6>…</H6>
```

该标记会自动换行，因此在不同标题之间不用加入换行标记。文字中通常存在不同位置，因此可加入 align=left/center/right 来设定。

【例 2-3】 在网页中分别设定标题一、二、三、四、五、六的标题大小(如图 2-3 所示)。

```
<HTML>
    <HEAD>
    <TITLE>一、二、三、四、五、六的标题大小</TITLE>
    </HEAD>
    <BODY>
    <H1 align="left"> 标题一 </H1>
    <H2 align="left"> 标题二 </H2>
    <H3 align="center"> 标题三 </H3>
    <H4 align="right"> 标题四 </H4>
    <H5> 标题五 </H5>
    <H6> 标题六 </H6>
    </BODY>
</HTML>
```

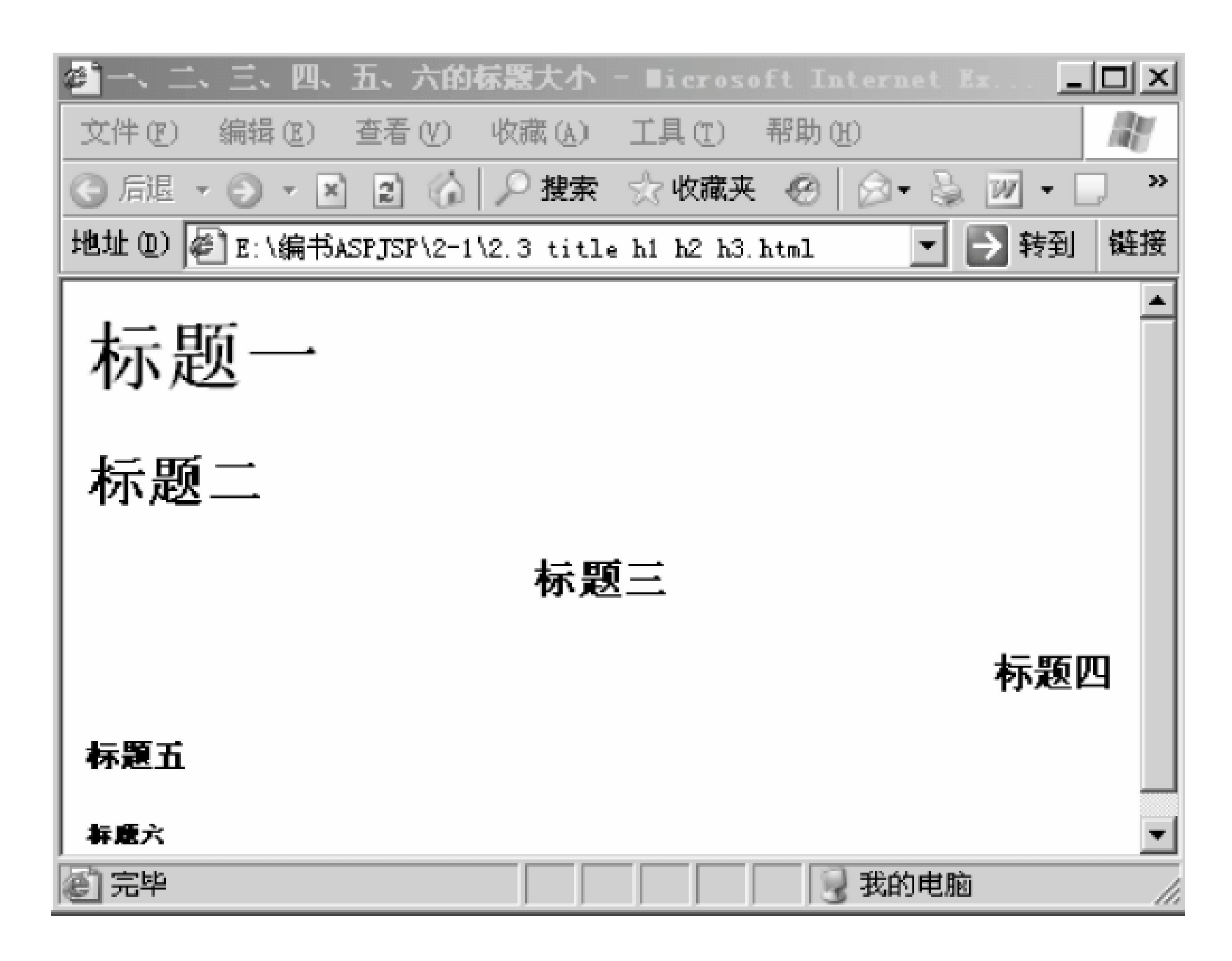


图 2-3 标题大小的显示效果

2.3.2 文本修饰标记

1. <FONT>基本字体标记

<FONT>标记指定选定文本的字体、大小和颜色。文本修饰标记的格式为：

```
<FONT size=n face=charnames color= #n>
```

注意： size 表示字体大小，face 表示字体类型，color 表示文本颜色。

【例 2-4】 设定不同字体、大小、类型、颜色(如图 2-4 所示)。

```
<HTML>
    <HEAD>
    <TITLE>字体大小类型颜色</TITLE>
    </HEAD>
    <BODY>
    <P>
```

```
    <FONT size="1" color="red" face="黑体">
        我爱你,中国!</FONT>
    </P>
    <P>
    <FONT size="3">欢迎光临!</H2>
    </P>
    <P>
    <FONT size="4" color="#037263" face="隶书">期待放假中!</H3>
    </P>
    </BODY>
</HTML>
```



图 2-4 font 的显示效果

注意：color 色彩值有两种设置方法：一种采用 RGB 色值；一种使用专有名词代表具体色彩，如 black(黑色)、olive(橄榄色)、teal(青色)、red(红色)、blue(蓝色)、maroon(莱色)、navy(浅蓝色)、gray(灰色)、silver(银色)、aque(浅绿色)、fuchsia(紫红色)、white(白色)、green(绿色)、purple(紫色)、yellow(黄色)。

2. 字体效果标记

字体效果标记包括粗体<B>、斜体<I>、下划线<U>。

强调文本标记为<EM>、<STRONG>，其中，<EM>相当于<I>，<STRONG>相当于<B>。

【例 2-5】 字体效果显示(如图 2-5 所示)。

```
<HTML>
    <HEAD>
    <TITLE>字体效果</TITLE>
    </HEAD>
    <BODY>
    <P>
    <B>粗体</B>
    </P>
    <P>
    <I>斜体</I>
    </P>
    <P>
    <U>下划线</U>
```

```
    </P>
    <P>
    <EM>斜体</EM>
    </P>
    <P>
    <STRONG>粗体</STRONG>
    </P>
    </BODY>
</HTML>
```



图 2-5　字体显示效果

2.3.3　段落格式标记

1. <P>标记

<P>标记产生一个新段落。

<P>标记的格式为：

```
<P  align = left/center/right  class = type>
```

2.
标记

标记产生段落换行。

标记的格式为：

```
<BR clear = none/left/right/all class = type>
```

3. <HR>标记

<HR>标记在网页窗口中加一条水平线，用以分隔文档。

<HR>标记的格式为：

```
<HR  align = left/center/right  size = n  width = n  color = #n>
```

项目案例 2：文本标记案例。

【题目】 制作学生名单显示页面，要出现标题、学生名单。

【代码】

```
<HTML>
    <HEAD>
    <TITLE>学生名单</TITLE>
    </HEAD>
    <BODY>
        2008 级学生名单
    <HR>
    <P>黄亮亮</P>
    <P>高燕</P>
    <P>王英姿</P>
    <P>王毅</P>
    <P>高轻若</P>
    </BODY>
</HTML>
```

2.4 实现超链接的标记码

超级链接又称为超文本链接，将文档中的对象与另外一个对象链接起来，这些对象可以是图片、文字、文档等元素。

超链接标记的格式为：

```
<A  class = type
     Id = value
     Href = reference
     Name = value
     Rel = same/next/parent/previous
     Rev = value
     Target = window
     Style = value
     Title = title
     Onclick = function
     Onmouseout = function
     Onmouseover = function>
</A>
```

项目案例 3：超链接标记案例。

【代码】

```
<HTML>
<HEAD>
    <TITLE>超链接</TITLE>
</HEAD>
<BODY>
```

```
    <P><A HREF = http://www.baidu.com target = "_blank" title = "百度首页">百度</A>
    <P><A HREF = "mailto:316932984@qq.com" >我的邮件地址</A>
    <P><A HREF = http://www.tup.tsinghua.edu.cn target = "_self" ><img src = "banner.jpg">百
度</A>
</BODY>
</HTML>
```

2.5 表单标记码

表单用来收集用户信息，并将这些信息提交给服务器处理。表单可以包含多种组件，比如标签、文本框、按钮、复选框、单选按钮等。

表单标记的格式为：

```
<FORM name = "表单名" action = "URL" method = "GET/POST">
</FORM>
```

action 指定将要接收表单数据的服务器端程序或动态网页的网址，这样才能将数据正确传给表单处理程序。

项目案例 4：教师学生登录注册案例。

【代码】

```
<HTML>
<HEAD>
<meta http-equiv = "Content-Type" content = "text/html; charset = gb2312">
<link href = "wangye9pt.css" rel = "stylesheet" type = "text/css">
<TITLE>用户信息</TITLE></HEAD>
<BODY>
<TABLE width = "200"  border = "0" align = "center" cellpadding = "0" cellspacing = "1" bgcolor =
"#FF6666" class = "w9pt">
  <TR align = "center" bgcolor = "#FFFFFF">
     <TD height = "30" colspan = "2"><SPAN class = "9ptred">用户信息
     </SPAN></TD>
  </TR>
  <TR bgcolor = "#FFFFFF">
     <TD align = "right"><SPAN class = "W12pt">
       <%
username = session("userREGID")
                 strsql = "select * From userinfo Where username = '"&username&"'"
                 set rs = db.execute(strsql)
                 %>
         </SPAN>登录 ID: </TD>
         <TD class = "9ptred"><% = rs("username") %></TD>
    </TR>
    <TR bgcolor = "#FFFFFF">
      <TD width = "48%" align = "right">注册身份: </TD>
      <TD width = "52%"><%
                 typedok = rs("typed")
                 if typedok = 1 then
```

```
            response.Write "领导"
            elseif typedok = 2 then
            response.Write "教师"
            else
            response.Write "学生"
            end if %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right">真实姓名：</TD>
  <TD><% = rs("REALNAME") %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right">登录密码：</TD>
  <TD><% = rs("PASSWORD") %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right">学号/教师编号：</TD>
  <TD><% = rs("num") %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right">E - mail：</TD>
  <TD><% = rs("email") %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right">联系电话：</TD>
  <TD><% = rs("tel") %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right">联系地址：</TD>
  <TD><% = rs("address") %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right">所在院系：</TD>
  <TD><% = rs("DEPART") %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right">所在班级：</TD>
  <TD><% = rs("CLASS1") %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right">注册时间：</TD>
  <TD><% = rs("submitdate") %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right">注册 IP：</TD>
  <TD><% = rs("IP") %></TD>
</TR>
<TR bgcolor = "#FFFFFF">
  <TD align = "right"> </TD>
  <TD height = "30"> </TD>
</TR>
```

```
    <TR bgcolor = "#FFFFFF">
      <TD colspan = "2" align = "center"><A HREF = "javascript:window.close();" class = "w9pt">【关
闭窗口】</A></TD>
    </TR>
</TABLE>
</BODY>
</HTML>
```

2.6 在 HTML 文档中使用 CSS 样式的方法

CSS 是 cascading style sheet 的缩写，是用于控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。

CSS 在引用中，必须首先在 head 中声明，然后在 body 中以 class＝样式的形式进行应用。

下面给出常用 CSS 样式的定义格式：

```
BODY {                          //文件体字体、大小、颜色、段间距样式设置
    COLOR: #000000;
    FONT: 12px 宋体;
    }
TD {
    COLOR: #000000;
    FONT: 12px 宋体;
    line-height:18px
}
TH {
    COLOR: #000000;
    FONT: 12px 宋体;
}
/*//Link Style//*/
a:link {                        //默认链接样式设置,访问前、悬浮中、点击、访问后样式设置
    COLOR: #ffffff;
    TEXT-DECORATION: none
}
a:visited {
    COLOR: #ffffff;
    TEXT-DECORATION: none
}
a:active {
    COLOR: #000000;
    TEXT-DECORATION: none
}
a:hover {
    COLOR: #ffffff;
    TEXT-DECORATION: underline
}
a.a1:visited {                  //A1 链接样式设置,访问前、悬浮中、点击、访问后样式设置
    COLOR: #ffffff;
```

```
    TEXT - DECORATION: none
}
a.a1:active {
    COLOR: #000000;
    TEXT - DECORATION: none
}
a.a1:hover {
    COLOR: #ffffff;
    TEXT - DECORATION: underline
}
/*//Link Style End//*/
.f1 {                          //F1 字体颜色、大小、样式设置
    COLOR: #ffffff;
    FONT: 14px 宋体;
      font - weight:bold;
}
```

项目案例 5：留学网站 CSS 标记使用案例。

【部分代码】

```
<HTML>
<HEAD>
<META http - equiv = "Content - Type" content = "text/html; charset = gb2312">
<TITLE>留学快讯</TITLE>
<META name = "GENERATOR" content = "Microsoft FrontPage 3.0">
<LINK rel = "stylesheet" href = "public/style.css">
</HEAD>
<BODY topmargin = "0" leftmargin = "0">
<DIV align = "center"><CENTER>
<TABLE
      border = "0" cellpadding = "0" cellspacing = "0" width = "717">
        <TR>
          <TD width = "24"><IMG src = "images/pic_002.jpg" width = "5" height = "24"></TD>
          <TD width = "59" class = "f2"><A HREF = "../../index.htm">留学全球</A></TD>
          <TD width = "10" class = "f2"><FONT color = "#FFFFFF">|</FONT></TD>
          <TD width = "73" class = "f2"><A HREF = "index.htm">加拿大首页</A></TD>
          <TD width = "12" class = "f2"><FONT color = "#FFFFFF">|</FONT></TD>
          <TD width = "64" class = "f2"><A HREF = "news.htm">留学快讯</A></TD>
          <TD width = "11" class = "f2"><FONT color = "#FFFFFF">|</FONT></TD>
          <TD width = "74" class = "f2"><A HREF = "index.htm">认识加拿大</A></TD>
          <TD width = "14" class = "f2"><FONT color = "#FFFFFF">|</FONT></TD>
          <TD width = "62" class = "f2"><A HREF = "college.htm">院校介绍</A></TD>
          <TD width = "12" class = "f2"><FONT color = "#FFFFFF">|</FONT></TD>
          <TD width = "60" class = "f2"><A HREF = "application.htm">申请指南</A></TD>
          <TD width = "316"></TD>
          <TD width = "16"><DIV align = "center"><CENTER><TABLE border = "0" cellpadding = "0"
          cellspacing = "0" width = "113%">
            <TR>
              <TD width = "63%"></TD>
              <TD width = "50%"><IMG src = "images/pic_003.jpg" width = "6" height = "24"></TD>
            </TR>
```

```
          </TABLE>
          </CENTER></DIV></TD>
        </TR>
      </TABLE>
      </CENTER></DIV></TD>
    </TR>
    <TR>
      <TD height="1" valign="top" width="715" bgcolor="#C8C8C8"><DIV align="center">
<CENTER>
</TABLE>
</BODY>
</HTML>
```

留学网站 CSS 标记显示效果如图 2-6 所示。



图 2-6 留学网的显示效果

2.7 本章小结

HTML 语言是网页专用语言，由不同的标记组成，是一种静态页面。

本章介绍了 HTML 标记语言的使用方法，同时给出了案例效果。在设计静态页面时，要注重界面色彩搭配以及常用标记的使用方法。

要求学过本章后，能够独自开发出静态页面，并且自行设计版式。

2.8 思考与练习

1. 什么是 HTML？
2. 一个静态网页页面由哪些内容构成？
3. 制作留学网页。
4. 制作网站的注册页面。

第3章 JavaScript脚本语言

JavaScript 脚本是众多脚本语言中较为优秀的一种，它与 WWW 的结合有效地实现了网络计算和网络计算机的蓝图，成为 Java 家族重要的一份子。本章详细介绍了 JavaScript 脚本语言的语法格式、校验用户名和密码、验证电子邮箱等。

学习目标及要求

(1) 了解 JavaScript 语言的运行原理及特点。

(2) 熟悉 JavaScript 的变量、数据类型、表达式的定义方法。

(3) 能够运用简单 JavaScript 语句编写程序。

(4) 了解 JavaScript 算术、日期时间、字符串、数组函数的使用。

(5) 了解鼠标驱动事件的处理方法。

3.1 JavaScript 语言简介

JavaScript 是由 Netscape 公司开发并随 Navigator 导航者一起发布的、介于 Java 与 HTML 之间、基于对象事件驱动的编程语言，正日益受到全球的关注。因它的开发环境简单，不需要 Java 编译器，而是直接运行在 Web 浏览器中，备受 Web 设计者喜爱。

JavaScript 的出现使得信息和用户之间不仅仅是一种显示和浏览的关系，而是实现了一种实时的、动态的、可交互式的表达能力。

3.1.1 JavaScript 的特点

为了运用 JavaScript 控制 HTML 页面上的对象，JavaScript 的代码必须与 HTML 代码结合在一起。将 JavaScript 嵌入 HTML 页面时，必须使用< SCRIPT >标签，该标签使用形式如下：

```
< SCRIPT Language = "JavaScript">
    //JavaScript 代码
</SCRIPT >
```

标签< SCRIPT >通知浏览器有脚本嵌入到标签中。

JavaScript 与 Java 的区别：虽然 JavaScript 与 Java 有紧密的联系，但却是两个公司开发的两个不同的产品。Java 是 Sun 公司推出的新一代面向对象的程序设计语言，特别适合

于 Internet 应用程序开发。JavaScript 是 Netscape 公司的产品，其目的是为了扩展 Netscape Navigator 功能而开发的一种可以嵌入 Web 页面中的基于对象和事件驱动的解释性语言，它的前身是 Live Script。Java 的前身是 Oak 语言。

3.1.2 JavaScript 的运行原理

Javascript 脚本可以运行在客户端，缓解因频繁访问对服务器造成的压力。比如我们验证用户名和密码，只须在本地验证就可以，如果用户名、密码正确，就通过验证，如果不正确，就不能登录主页，如图 3-1 所示。

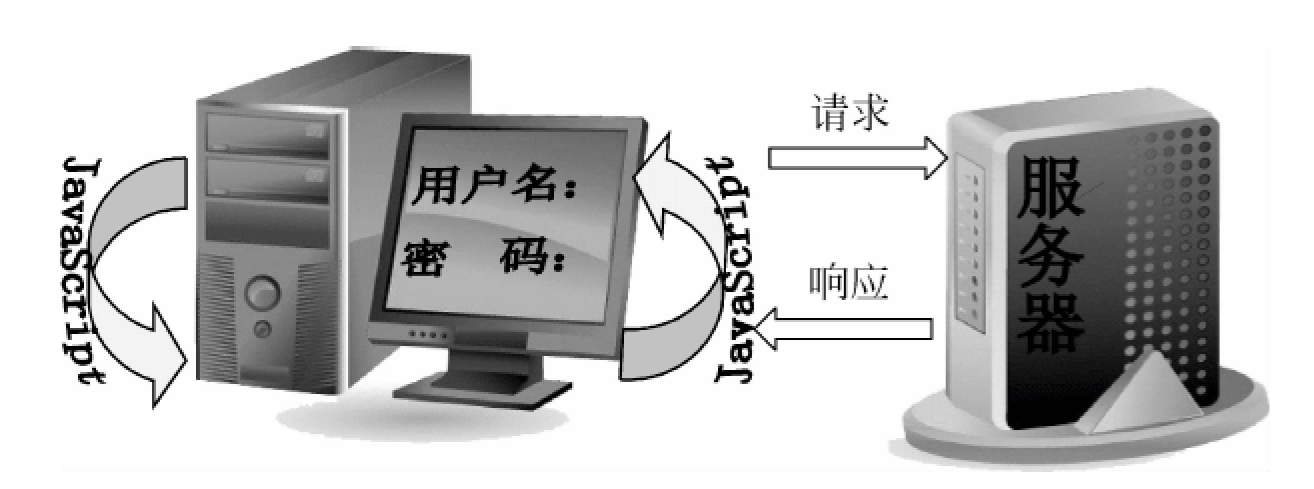


图 3-1 JavaScript 的运行原理

项目案例 1：编写第一个 JavaScript 程序(如图 3-2 所示)。

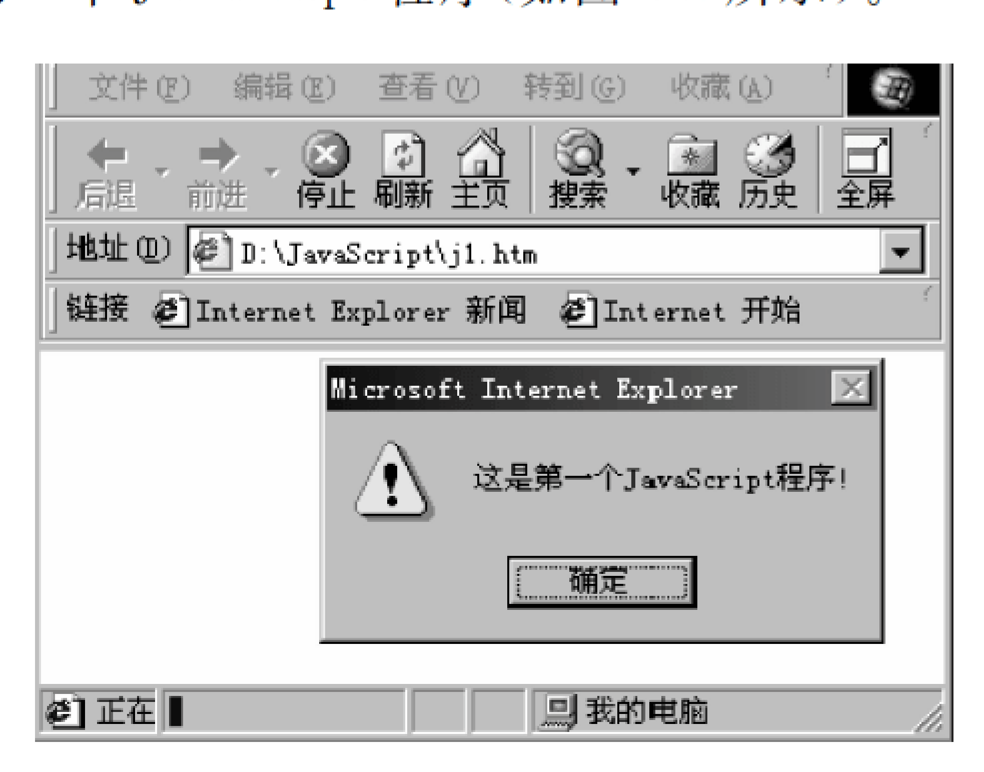


图 3-2 第一个 JavaScript 程序

【题目】 使用 JavaScript 代码，输出“欢迎进入 JavaScript 世界!”。

【代码】

```
<HTML>
<HEAD>
<SCRIPT Language = "JavaScript">
   //JavaScript
alert("这是第一个 JavaScript 程序!");
alert("欢迎进入 JavaScript 世界!");
alert("今后我们将共同学习 JavaScript 知识!");
</SCRIPT>
</HEAD>
</HTML>
```

解释：

JavaScript 代码由 <SCRIPT Language ="JavaScript">...</SCRIPT>说明。在标识<SCRIPT Language ="JavaScript">...</SCRIPT>之间就可加入 JavaScript 脚本。

alert( )是 JavaScript 的窗口对象方法，其功能是弹出一个 OK 对话框并显示()中的字符串。

通过<! —— ...//——>标识说明，若不认识 JavaScript 代码的浏览器，则所有在其中的标识均被忽略；若认识，则执行其结果。使用注释是一个好的编程习惯，它使其他人可以读懂你的语言。

JavaScript 以</SCRIPT>标签结束。

JavaScript 中注释代码的方法：单行注释用//，多行注释用/ * … * /。

3.2 JavaScript 变量、数据类型与表达式

3.2.1 JavaScript 变量和数据类型

JavaScript 中采用弱类型变量，变量可以不做声明和不做类型说明，而在使用或赋值时确定类型。但为了形成良好的编程风格，变量应该采取先定义、再使用的方法。JavaScript 中变量的定义用关键字 var 来实现。例如，定义一个名为 id 的变量：

```
<SCRIPT Language = "JavaScript">
    var id;
</SCRIPT>
```

备注：强类型变量和弱类型变量的对比。

Java 采用强类型变量检查，即所有变量在编译之前必须做声明。例如：

```
Integer x;
String y;
x = 1234;
y = 4321;
```

其中，x=1234 说明是一个整数，y=4321 说明是一个字符串。

JavaScript 中变量声明采用弱类型，即变量在使用前无须做声明，而是解释器在运行时检查其数据类型，如：

```
x = 1234;
y = "4321";
```

其中，前者说明 x 为其数值型变量，后者说明 y 为字符型变量。

在 JavaScript 中有 4 种基本的数据类型：数值型(整数和实数)、字符串型(用“ ”或‘ ’括起来的字符或数值)、布尔型(用 True 或 False 表示)和空值。

变量的命名要遵循以下规则：

(1) 变量名不能与保留字冲突。

(2) 变量名必须以字母或者下划线(_)开头,不能以数字或者其他非字母字符作为变量名开头。

(3) 变量名中不能包含空格。

(4) JavaScript 是区分大小写的,所以给变量命名时要考虑大小写的问题。

3.2.2 JavaScript 表达式

在定义完变量后,就可以对它们进行赋值、改变、计算等一系列操作。表达式是指具有一定的值,用运算把常数和变量连接起来的代数式。可以说它是变量、常量、布尔及运算符的集合,因此表达式可以分为算术表述式、字符串表达式、赋值表达式以及布尔表达式等。

JavaScript 中的算术运算符有单目运算符和双目运算符。

双目运算符:

+(加)、-(减)、*(乘)、/(除)、%(取模)、|(按位或)、&(按位与)、<<(左移)、>>(右移)、>>>(右移,零填充)

单目运算符:

-(取反)、~(取补)、++(递加 1)、--(递减 1)

比较运算符首先对操作数进行比较,再返回一个 True 或 False 值,有 8 个比较运算符:

<(小于)、>(大于)、<=(小于等于)、>=(大于等于)、=(等于)、!=(不等于)

在 JavaScript 中增加了几个布尔逻辑运算符:

!(取反)、&=(与之后赋值)、&(逻辑与)、|=(或之后赋值)、|(逻辑或)、^=(异或之后赋值)、^(逻辑异或)、?:(三目操作符)、||(或)、==(等于)、|=(不等于)

其中,三目操作符主要格式如下:

操作数? 结果 1:结果 2

若操作数的结果为真,则表达式的结果为结果 1,否则为结果 2。

项目案例 2:编写文字跟随鼠标移动代码。

【题目】 使用 JavaScript 代码,实现文字跟随鼠标移动效果(如图 3-3 所示)。



图 3-3 文字跟随鼠标移动效果

【代码】

```
<SCRIPT Language = "JavaScript">
<! -- Begin
msg = 'My Future at My Fingertips!';
```

```
font = 'Verdana,Arial';
size = 2; //1 - 7 only!
color = '#016E30';
speed = 0.5;                    //移动步数
msg = msg.split('');            //将字符串 msg 分割成字符串数组 msg
n = msg.length;
a = size * 5;                   //定义每个字符的显示宽度
ymouse = 0;
xmouse = 0;
props = "<font face = " + font + " size = " + size + " color = " + color + ">";
document.write('<div id = "outer" style = "position:absolute;top:0px;left:0px"><div style =
"position:relative">');      //为显示信息的每个字符定义一个层,都位于左上角
for (i = 0; i < n; i ++ )
document.write('<divid = "iemsg" style = "position:absolute;top:0px;left:0;height:' + a + ';
width:' + a + ';text - align:center">' + props + msg[i] + '</font></div>');
document.write('</div></div>');
function Mouse(){               //获取当前鼠标位置,并在 x 左边增加 20,生成显示信息的起始位置
ymouse  =  event.y;
xmouse  =  event.x + 20;
}
document.onmousemove = Mouse; //定义当前文档的鼠标移动事件 Mouse
Y = new Array();
X = new Array();
Y = new Array();
X = new Array();
for (i = 0; i < n; i ++ ){
Y[i] = 0;
X[i] = 0;
Y[i] = 0;
X[i] = 0;
}
</SCRIPT>
```

3.3 JavaScript 的选择语句

3.3.1 if 语句的用法

1. if 语句格式

```
if (表达式)
  {
   语句块;
  }
```

功能：若表达式为 true,则执行语句块。

2. if-else 语句格式

```
if (表达式)
  {
   语句块 1;
   }
else
  {
   语句块 2;
   }
```

功能：若表达式为 true,则执行语句块 1；否则执行语句块 2。

3. if-else 语句格式

```
if(表达式)语句 1;
else(表达式)语句 2;
else if(表达式)语句 3;
⋮
else 语句 4;
```

功能：在这种情况下,每一级的布尔表达式都会被计算,若为真,则执行其相应的语句,否则执行 else 后的语句。

项目案例 3：if 条件判断语句 1。

【题目】 填写学生信息并提交。学生信息不允许为空,若提交无误则显示提交的学生信息,否则提示错误信息(如图 3-4 所示)。

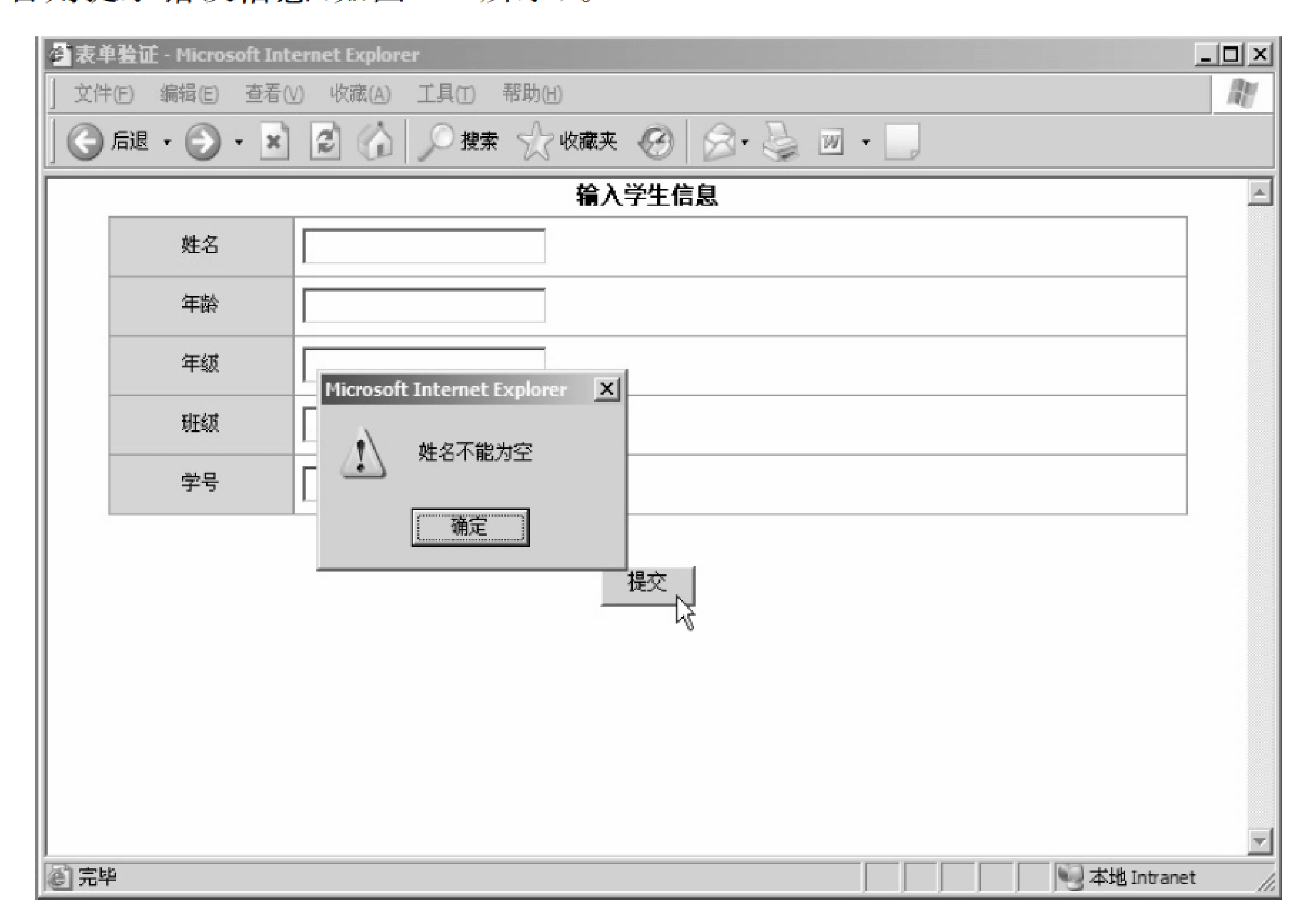


图 3-4 输入学生信息并判断

【代码】

```
function checkNull(txtItem,txtName){
  If (txtItem.value == "")              //若文本框对象的值为空
  {
    alert(txtName + "不能为空");     //进行提醒
    txtItem.focus();                 //将光标焦点指向该文本框
    Return false;                    //函数返回值为假

  }Else Return true;                 //否则表示填写过内容,函数返回值为真
}
Function checkForm(){
If (checkNull(document.getElementById("name"),"姓名")&&
    checkNull(document.getElementById("age"),"年龄")&&
    checkNull(document.getElementById("grade"),"年级")&&
    checkNull(document.getElementById("stuClass"),"班级")&&
    checkNull(document.getElementById("stuNo"),"学号"))
Return true;                         //若以上判断均为真,返回真
Else Return false;                   //否则返回假
}
```

项目案例 4：if 条件判断语句 2。

【题目】 在客户端验证 E-mail 格式是否正确。

【代码】

```
<SCRIPT Language = "JavaScript">
function check(myform){
    if(myform.e_mail.value == ""){
        alert("请输入 E-mail 地址!");myform.e_mail.focus();return;
   }
    if(!checkemail(myform.e_mail.value)){
        alert("您输入的 E-mail 地址不正确!");myform.e_mail.focus();return;
   }
    myform.submit();
}
function checkemail(email){
    var str = email;
     //在 JavaScript 中,正则表达式只能使用"/"开头和结束,不能使用双引号
    var Expression = /\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/;
    var objExp = new RegExp(Expression);
    if(objExp.test(str) == true){
        return true;
    }else{
        return false;
    }
}
</SCRIPT>
```

3.3.2 switch 语句的用法

```
switch (表达式)
 {case 值 1: 语句 1;
```

```
        break;
    case 值 2: 语句 2;
        break;
        ⋮
    case 值 n: 语句 n;
        break;
    default: 语句;
    }
```

功能：判断表达式属于 case 的哪一个值，然后依据对应的 case 值执行相应的语句。

项目案例 5：编写 JavaScript 代码。

【题目】 使用 JavaScript 代码，实现提示文字的变色效果（如图 3-5 所示）。



图 3-5 变色文字

【代码】

```
<SCRIPT Language = "JavaScript1.2">
var message = " 招聘启事"
var neonbasecolor = "#FF930D"
var neontextcolor = "#00B3FD"
var flashspeed = 300                    //设置变色速度
var n = 0
if (document.all){
document.write('<font color = "' + neonbasecolor + '">')
for (m = 0;m < message.length;m ++ )
document.write('<span id = "neonlight" class = "content">' + message.charAt(m) + '</span>')
document.write('</font>')
var tempref = document.all.neonlight //cache reference to neonlight array
}
else
document.write(message)
function neon(){                        //改变字母基本色
if (n == 0){
for (m = 0;m < message.length;m ++ )
tempref[m].style.color = neonbasecolor
}   //循环改变字母的色彩
tempref[n].style.color = neontextcolor
if (n < tempref.length - 1)
n ++
else{
n = 0
clearInterval(flashing)
setTimeout("beginneon()",1500)
return
} }
```

```
function beginneon(){
if (document.all)
flashing = setInterval("neon()",flashspeed)}
beginneon()
</SCRIPT>
```

3.4　JavaScript 的循环语句

3.4.1　for 循环语句

基本格式为：

```
for(初始化; 条件; 增量)
    语句集;
```

功能：实现条件循环，当条件成立时，执行语句集，否则跳出循环体。

说明：

- 初始化：告诉循环的开始位置，必须赋予变量的初值。
- 条件：是用于判别循环停止时的条件。若条件满足，则执行循环体，否则，跳出。
- 增量：主要定义循环控制变量在每次循环时按什么方式变化。

项目案例 6：for 循环语句案例 1。

【题目】　判断单选框对象选中案例（如图 3-6 所示）。

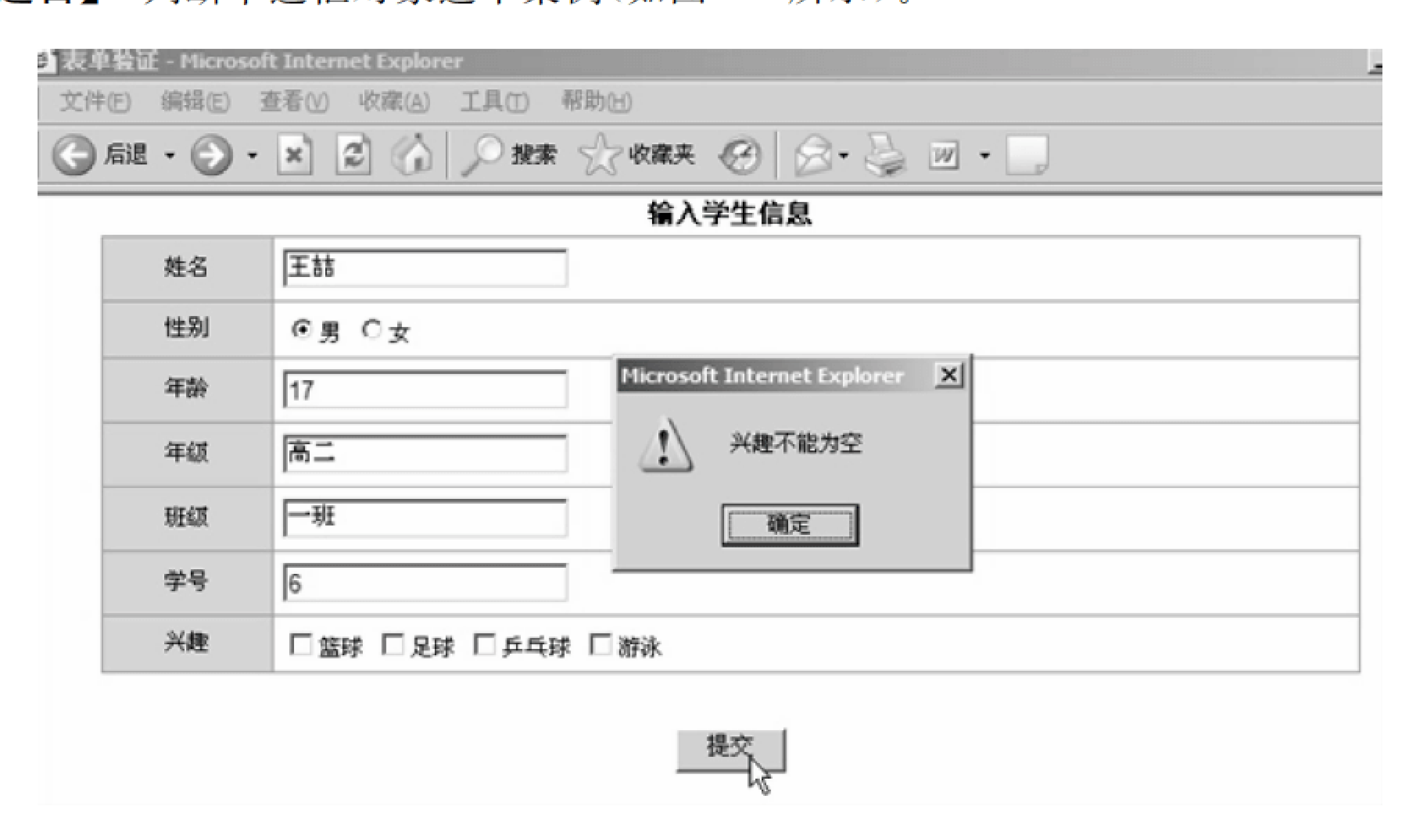


图 3-6　判断单选框和复选框是否被选中

【代码】

```
Function checkItem(itemObj,itemName){
  var flag = false;
  For(var i = 0;i < itemObj.length;i ++ ){        //循环检查单选框对象是否被选中
    If(itemObj[i].checked){
```

```
      flag = true;}                          //做标记
  }
  If(flag)                                   //若被选中
    Return true;
  Else{                                      //未被选中
   alert(itemName + "不能为空");
     //提示
     Return false;          }  }
  checkItem(document.all.sex,"性别")          //检查单选框是否被选中
  checkItem(document.all.interest,"兴趣"))    //检查复选框是否被选中
```

项目案例 7：for 循环语句案例 2。

【题目】 循环判断复选框选中案例(如图 3-7 所示)。

图 3-7 循环检查复选框对象是否被选中

【代码】

```
Function checkAll(name)
{
    var el = document.getElementsByTagName('input');
    var len = el.length;                      //读取元素的个数
        For(var I = 0; I < len; I ++ )        //循环元素个数的次数
    {
        If((el[I].type == “checkbox”) && (el[i].name == name)) {
            If(document.getElementById(“ca”).checked) {
                el[i].checked = true;
                        }
          Else{//用户取消全选
            el[I].checked = false; }} } }
```

3.4.2 while 循环语句

基本格式为：

```
while(条件)
语句集;
```

该语句与For语句一样,当条件为真时,重复循环,否则退出循环。

For与while语句的比较:两种语句都是循环语句,使用For语句在处理有关数字时更易看懂,也较紧凑;而while循环对复杂的语句效果更特别。

3.4.3 do-while循环语句

基本格式为:

```
do{  语句块;
    计数器表达式;
  }while (循环条件表达式)
```

注意:do-while语句和while语句的最大区别是:do-while语句先执行语句块,后执行条件判断;while语句先执行条件判断,后执行语句块。二者执行的次数不一样。

3.5 JavaScript的函数

JavaScript中的函数不同于其他语言的函数,每个函数都是作为一个对象被维护和运行的。通过函数对象的性质,可以很方便地将一个函数赋值给一个变量或者将函数作为参数传递。

JavaScript不区分函数和过程,它只有函数。

(1) 函数的定义:

```
founction 函数名([参数1,参数2,…])
 { 代码块; }
```

(2) 函数的参数:在函数定义时确定参数,然后按照确定的参数进行传递调用。

(3) 函数返回值:可以使用return语句返回常量、变量,也可以是表达式等。

在函数的定义中,我们看到函数名后有参数表,这些参数变量可能是一个或几个。那么怎样才能确定参数变量的个数呢?在JavaScript中可通过arguments.Length来检查参数的个数。

例如:

```
Function function_Name(exp1,exp2,exp3,exp4)
Number = function _Name . arguments .length;
if (Number>1)
document.wrile(exp2);
if (Number>2)
document.write(exp3);
if(Number>3)
document.write(exp4);
…
```

3.5.1 算术函数

Math 对象无须用 new 操作符创建对象，而是可以直接使用，所以又被称做静态的对象。

调用方式为：

```
Math.数学函数名(参数表)
```

1. Math 对象的属性

Math 中提供了 6 个属性，它们是数学中经常用到的常数 E、以 10 为底的自然对数 ln10、以 2 为底的自然对数 ln2、3.14159 的 PI、1/2 的平方根 SQRT(1/2)、2 的平方根 SQRT(2)。

2. Math 对象的方法

功能：提供除加、减、乘、除以外的运算，如对数、平方根等。

主要方法如下。

- 绝对值：abs()。
- 正弦、余弦：sin()、cos()。
- 反正弦、反余弦：asin()、acos()。
- 正切、反正切：tan()、atan()。
- 四舍五入：round()。
- 平方根：sqrt()。
- base 的 exponent 次幂：pow(base，exponent)。

3.5.2 String 函数

String 中的函数不能直接使用类名 String 加函数名，而是要创建 String 类型的对象，在对象的后面加上函数名。

1. String 对象的定义

```
str1 = "hello";
str2 = new String("hello");
```

2. String 对象的属性

String 对象只有一个属性，即 length。它表明了字符串中的字符个数，包括所有符号。

例如：

```
mytest = "This is a JavaScript"
mystringlength = mytest.length
```

最后，mystringlength 返回 mytest 字符串的长度为 20。

3. String 对象的方法

String 对象的方法共有 19 个，主要用于有关字符串在 Web 页面中的显示、字体大小、字体颜色、字符的搜索以及字符的大小写转换。

3.5.3 日期及时间

JavaScript 中没有日期类型，Date 对象中不仅包括日期，还包括时间。

功能：提供一个有关日期和时间的对象。

使用 New 运算符创建一个实例。例如：

```
MyDate = New Date()
```

Date 对象没有提供直接访问的属性，只具有获取和设置日期和时间的方法。

获取日期和时间的方法如下。

- getYear()：返回年数。
- getMonth()：返回月份数。
- getDate()：返回日期数。
- getDay()：返回星期几。
- getHours()：返回小时数。
- getMinutes()：返回分钟数。
- getSeconds()：返回秒数。
- getTime()：返回毫秒数。

设置日期和时间的方法如下。

- setYear()：设置年。
- setMonth()：设置月份数。
- setDate()：设置日期数。
- setDay：设置星期几。
- setHours()：设置小时数。
- setMinutes()：设置分钟数。
- setSeconds()：设置秒数。
- setTime()：设置毫秒数。

3.5.4 Array 对象

数组是一系列元素的有序集合，它的强大功能是不可替代的。在 JavaScript 中，可以使用 Array(数组)对象来完成对数组的操作。

1. 定义对象的数组

```
Function arrayName(size){
This.length = Size;
for(var X = ; X <= size;X ++ )
this[X] = 0;
```

```
Reture this;
}
```

其中，arrayName 是定义数组的一个名字，Size 是有关数组大小的值(1-size)，即数组元素的个数。通过 for 循环对一个当前对象的数组进行定义，最后返回这个数组。

2. 创建数组实例

一个数组定义完成以后，还不能马上使用，必须为该数组创建一个数组实例：

```
Myarray = New arrayName(n);
```

并赋予初值：

```
Myarray[1] = "字串 1";
Myarray[2] = "字串 2";
Myarray[3] = "字串 3";
⋮
Myarray[n] = "字串 n";
```

一旦给数组赋予初值后，数组中就具有真正意义的数据了，以后就可以在程序设计过程中直接引用。

项目案例 8：函数案例 1。

【题目】 编写一个用户选择头像的功能，将头像名称存放在下拉列表中，当用户选择某个头像后，可以在页面中将这个头像的图片显示出来(如图 3-8 所示)。





单击图片名称　　显示对应头像

图 3-8 用户选择头像和用户显示图像

【代码】

```
<SCRIPT Language = "javascript">
Function chgFace() {
  document.forms[0].submit();
}
</SCRIPT>
<FORM action = "" method = "post">
```

```
⋮
<SELECT name="userFace" onchange="chgFace();">
⋮
<IMG src="../images/userface/<%=Request("userFace")%>">
⋮
</FORM>
```

项目案例 9：函数案例 2。

【题目】 检查表单元素是否为空。

【代码】

```
Index.asp:
  <!--#include file="function.asp"-->
  <HTML>
  <HEAD>
  <TITLE>检查表单元素是否为空</TITLE>
  <META http-equiv="Content-Type" content="text/html; charset=gb2312">
  <LINK href="style.css" rel="stylesheet">
  </HEAD>

  <BODY><form name="form1" method="post" action="">
<TABLE width="557" height="121"  border="0" cellpadding="0" cellspacing="0" class="
tab_border">
    <TR>
      <TD valign="top" background="Images/top.jpg"> </TD>
  </TR>
  </TABLE>
  <TABLE width="557" height="129"  border="0" cellpadding="0" cellspacing="0">
    <TR>
    <TD align="center" valign="top"><TABLE width="90%" height="202"  border="0"
cellpadding="0" cellspacing="0" class="tab_border">
    <TR>
        <TD width="17%" height="30" align="center">留 言 人: </TD>
        <TD width="83%"><INPUT type="text" name="mname" size="20"></TD>
    </TR>
    <TR>
        <TD height="30" align="center">留言主题: </TD>
        <TD><input type="text" name="mtext" size="66"></TD>
    </TR>
    <TR>
        <TD align="center">留言内容: </TD>
        <TD><textarea name="mnote" cols="64" rows="10" class="wenbenkuang"></textarea>
</TD>
    </TR>
    <TR align="center">
        <TD height="4" colspan="2"><INPUT name="Button" type="button" class="btn_
grey" value="提交" onClick="check(form1)">
            
           <input name="Submit2" type="reset" class="btn_grey" value="重置"></TD>
        </TR>
    </TABLE></TD>
      </TR>
    </TABLE>
    </FORM>
```

```
    </BODY>
    </HTML>
  Function.asp 文件:
  <SCRIPT Language="javascript">
  //检查表单元素是否为空
  function check(Form){
    for(i=0;i<Form.length;i++){
          if(Form.elements[i].value == ""){ //Form 的属性 elements 的首字 e 要小写
            alert(Form.elements[i].name + "不能为空!");
            Form.elements[i].focus();
            return;
          }
      }
      Form.submit();
}
</SCRIPT>
```

3.6 事件驱动及事件处理

JavaScript 是基于对象(object-based)的语言。这与 Java 不同,Java 是面向对象的语言。基于对象的基本特征就是采用事件驱动(event-driven)。它是在界面的环境下,使得一切输入变得简单化。通常将鼠标或热键的动作称为事件(event),而由鼠标或热键引发的一连串程序的动作称为事件驱动(event driver)。对事件进行处理的程序或函数称为事件处理程序(event handler)。

在 JavaScript 中对象事件的处理通常由函数(function)担任。基本格式与函数的使用一样,可以将前面所介绍的所有函数作为事件处理程序。

格式如下:

```
Function 事件处理名(参数表){
事件处理语句集;
 ⋮
}
```

主要有以下几个事件。

- 单击事件: onClick。
- 改变事件: onChange。
- 选中事件: onSelect。
- 获得焦点事件: onFocus。
- 失去焦点: onBlur。
- 载入文件: onLoad。
- 卸载文件: onUnload。

项目案例 10: 事件处理案例 1。

【代码】

```
<HTML>
<HEAD>
<SCRIPT Language="JavaScript">
```

```
<! --
function loadform(){
alert("这是一个自动装载例子!");
}
function unloadform(){
alert("这是一个卸载例子!");
}
// -- >
</SCRIPT >
</HEAD >
< BODY OnLoad = "loadform()" OnUnload = "unloadform()">
< A HREF = "test.htm">调用</A >
</BODY >
</HTML >
```

项目案例 11：事件处理案例 2(如图 3-9 所示)。

【代码】

```
< HTML >
< HEAD >
< SCRIPT Language = "JavaScript"><! --
// -- >
function makeArray(n){
this.length = n
return this
}
function hexfromdec(num) {
hex = new makeArray(1);
var hexstring = "";
var shifthex = 16;
var temp1 = num;
for(x = 1; x >= 0; x -- ) {
hex[x] = Math.round(temp1/shifthex - .5);
hex[x - 1] = temp1 - hex[x] * shifthex;
temp1 = hex[x - 1];
shifthex / = 16;
}
for (x = 1; x>= 0; x -- ) { hexstring += getletter(hex[x]); }
return (hexstring);
}
function getletter(num) {
if (num < 10) { return num; }
else {
if (num == 10) { return "A" }
if (num == 11) { return "B" }
if (num == 12) { return "C" }
if (num == 13) { return "D" }
if (num == 14) { return "E" }
if (num == 15) { return "F" }
}
}
function rainbow(text){
var color_d1;
var allstring = "";
```

```
for(i = 0;i < text.length;i = i + 2){
color_d1 = 255 * Math.sin(i/(text.length/3));
color_h1 = hexfromdec(color_d1);
allstring += "< FONT COLOR = " + color_h1 + "ff" + color_h1 + ">" + text.substring(i,i + 2) +
"</FONT >";
}
return allstring;
}
function sizefont(text){
var color_d1;
var allstring = "";
var flag = 0;
for(i = 0,j = 0;i < text.length;i = i + 1){
if (flag == 0) {
j ++ ;
if (j >= 7) {
flag = 1;}}
if (flag == 1) {
j = j - 1;
if (j <= 0) {
flag = 0; }}
allstring += "< FONT SIZE = " + j + ">" + text.substring(i,i + 1) + "</FONT >";
}
Return allstring;
}
document.write("< font size = 8 >< CENTER >")
document.write("< BR >< BR >")
document.write( sizefont("这是一个获取 Web 浏览器的程序"))
document.write("</CENTER ></FONT >")
document.write("浏览器名称: " + navigator.appName + "< BR >");
document.write("版本号: " + navigator.appVersion + "< BR >");
document.write("代码名字: " + navigator.appCodeName + "< BR >");
document.write("用户代理标识: " + navigator.userAgent);
</SCRIPT >
< BODY >
</BODY >
</HTML >
```



图 3-9 获取 Web 浏览器的程序

3.7 JavaScript 应用实例——JavaScript 表单验证

假如开发了一个网站，一天有 50 名用户注册使用。当浏览数据录入项时，发现用户提供的大部分信息是错误的，例如，电子邮件地址是“123.com”，电话号码是“addefgr”等。问题在于用户提交注册信息之前，忽略了表单验证这个中间步骤。

大多数要求用户输入的网站，不管是针对简单的邮件列表还是更复杂的购物系统，都使用表单验证来减少系统数据库的错误数据。因此，程序表单验证可以帮助识别所输入数据类型的真实性，减少错误数据进入数据库。

表单验证是 JavaScript 最常见的用途之一。对于检查用户输入是否存在错误和是否疏漏了要求的字段，JavaScript 是一种十分便捷的方法。正如在客户端脚本中所讲的，这有助于减少往返服务器的次数。

在以下示例中，如发生下列情况，表单验证程序将显示警告消息：

(1) 字段“名称”留为空白。

(2) 性别未选定。

(3) 输入的密码少于 6 个字符。

(4) 指定的电子邮件中没有字符“@”。

(5) 年龄不在 1～99 的范围内，或者留为空白。

```
<HTML>
<HEAD>
<TITLE>表单验证</TITLE>
<SCRIPT LANGUAGE = \"JavaScript\">
<! --
function validate()
{
   f = document.reg_form;
   if(f.uname.value == \"\")
   {
     alert(\"输入姓名\");
     f.uname.focus();
     return false;
   }
   if(f.gender[0].checked == false&&f.gender[1].checked == false)
   {
     alert(\"请指定性别\");
     f.gender[0].focus();
     return false;
   }
   if((f.password.value.length < 6)||(f.password.value == \"\"))
   {
     alert(\"请输入至少有 6 个字符的密码!\");
     f.password.focus();
     return false;
   }
```

```
    q = f.email.value.indexOf(\"@\")
    if(q == -1)
    {
      alert(\"请输入有效的电子邮件地址\");
      f.email.focus();
      return false;
    }
    if(f.age.value<1||f.age.value>99||isNaN(f.age.value))
    {
      alert(\"请输入有效的年龄!\");
      f.age.focus();
      return false;
    } [Page]
}
//-->
</SCRIPT>
</HEAD>
<BODY onLoad = \"document.reg_form.uname.focus()\" bgColor = \"limegreen\">
<FORM NAME = \"reg_form\" onSubmit = \"return validate()\" action = \"submit.htm\">
<CENTER>
<H1><U><FONT color = \"yellow\">欢迎来到 Aptech 网上家园</FONT></U><H1>
姓名:
<BR>
<INPUT TYPE = \"text\" NAME = \"uname\">
<P>
性别:<BR>
<INPUT TYPE = \"radio\" NAME = \"gender\" VALUE = \"男\">男
<INPUT TYPE = \"radio\" NAME = \"gender\" VALUE = \"女\">女
<BR><BR>
密码:<BR>
<INPUT TYPE = \"password\" NAME = \"password\" ID = \"password\">
<P>
电子邮件地址:<BR>
<INPUT TYPE = \"text\" NAME = \"email\" ID = \"email\">
<P>
年龄:<BR>
<INPUT TYPE = \"text\" NAME = \"age\">
<P>
<INPUT TYPE = \"submit\" NAME = \"submit\" VALUE = \"注册\">
</CENTER>
</FORM>
</BODY>
</HTML>
```

在浏览器中打开时,注意以上示例的电子邮件地址验证,检查是否包含“@”字符,在复杂的验证程序中,一般还需要通过检查空格和特殊字符来验证电子邮件地址的合法性。

总结:

(1) 单选按钮对象用于一组相互排斥的值,也就是用户只能从选项列表中选择一项。

(2) 单选按钮对象的 checked 属性可用于检查选项按钮是否被选中。

(3) 与单选按钮对象相关联的事件处理程序是 onBlur、onClick 和 onFocus。

(4) 列表框中可选择的项目是使用<OPTION>标记在<SELECT>和</SELECT>标记之间定义的。需要使用 SELECT 对象和 OPTION 对象进行控制。

(5) 与列表框对象相关联的事件处理程序是 onBlur、onChange 和 onFocus。

项目案例 12：综合 JavaScript 实例 1。

【题目】 实际项目开发中，大部分情况下是根据网页的特殊性，比如头部页面和尾部页面相同，将网页制作成 3 个页面：top.htm(头部页面)、bottom.htm(尾部页面)、index.htm(主内容页面)，然后将 top.htm、bottom.htm 两个页面全部链接到 index.htm 页面中，通过 JavaScript 代码实现，如图 3-10 所示。



图 3-10 合成后完整的 index.htm 页面

【代码】

将 top.htm 文件加载到 index.htm 的位置(如图 3-11 所示)：

```
<TABLE border = "0" cellpadding = "0" cellspacing = "0" width = "715">
    <TR>
        <TD width = "715"><SPAN id = "showImport"></SPAN><IE:Download ID = "oDownload" STYLE =
"behavior:url(#default#download)" /><SCRIPT>
function onDownloadDone(downDate){
showImport.innerHTML = downDate
}
oDownload.startDownload('top.htm',onDownloadDone)
</SCRIPT></TD>
    </TR>
    <TR>
        <TD width = "715"></TD>
    </TR>
</TABLE>
```



图 3-11 top. htm 的页面

将 bottom. htm 文件加载到 index. htm 的位置(如图 3-12 所示):

```
<TABLE border = "0" cellpadding = "0" cellspacing = "0" width = "715">
    <TR>
        <TD width = "715" height = "3"><SPAN id = "showImport1"></SPAN><IE:Download ID =
"oDownload1" STYLE = "behavior:url(#default#download)" /><SCRIPT>
function onDownloadDone1(downDate){
showImport1.innerHTML = downDate
}
oDownload1.startDownload('bottom.htm',onDownloadDone1)
</SCRIPT></TD>
    </TR>
</TABLE>
```

\界面设计\wz36920040410\canada\gb\bottom.htm

ENGLISH 政策法规 常用链接 寻求合作 广告服务 友情链接 版权声明 隐私权保护

Copyright 2004 Overseasstudy.cn All Rights Reserved

图 3-12 bottom. htm 的页面

项目案例 13: 综合 JavaScript 实例 2。

【题目】 网站注册页面用 JavaScript 代码实现。

【代码】

```
<HTML xmlns = "http://www.w3.org/1999/xhtml">
<HEAD>
<TITLE>注册页</TITLE>
<LINK href = "../CSS/style.css" rel = "stylesheet" type = "text/css" />
<STYLE>
/* 细线表格样式 */
/* 加到表格上的样式 */
.table_Tb {
   WIDTH: 70%;
   BORDER-COLLAPSE: collapse;
   border:1px #999999 solid;
}
/* 标题单元格样式 */
.TdTitle {
  BORDER: #999999 1px solid;
```

```
    padding:5px;
    BACKGROUND - COLOR: #d7d4cc;
    TEXT - ALIGN: center;
  border - top: 1px solid #999999;
  border - right: 1px solid #999999;
  border - bottom: 1px solid #999999;
  border - left: 1px solid #999999;
}
/* 内容单元格样式 */
.TdContent {
    BORDER: #999999 1px solid;
    padding: 5px;
    BACKGROUND - COLOR: #FFFFFF;
    TEXT - ALIGN: LEFT;

  border - top: 1px solid #999999;
  border - right: 1px solid #999999;
  border - bottom: 1px solid #999999;
  border - left: 1px solid #999999;
}
</STYLE>
<script language = "javascript">
function checkNull(txtItem, txtName){                //参数分别表示文本框对象及其对应中文
    if (txtItem.value == "")                         //若文本框对象的值为空
{
    alert(txtName + "不能为空");                     //进行提醒
    txtItem.focus();                                 //将光标焦点指向该文本框
    return false;                                    //函数返回值为假
    }else return true;                               //否则表示填写过内容,函数返回值为真
}
function checkItem(itemObj, itemName){               //参数分别表示单选框对象及其对应中文
    var flag = false;
    for(var i = 0;i < itemObj.length;i ++ ){         //循环检查单选框对象是否被选中
      if(itemObj[i].checked){
        flag = true;}                                //做标记
    }
    if(flag)                                         //若被选中
      return true;
    else{                                            //未被选中
      alert(itemName + "不能为空");                  //提示
      return false;
      }
}
function checkForm(){                                //验证表单元素是否为空
      if (
        checkNull(document.getElementById("uid"),"用户名")&&
        checkNull(document.getElementById("pwd"),"密码")&&
        checkItem(document.all.sex,"性别")&&
        checkNull(document.getElementById("userFace"),"头像")&&
        checkNull(document.getElementById("email"),"电子邮件")&&
        checkItem(document.all.character,"性格")&&
```

```
        checkItem(document.all.interest,"兴趣爱好")
      )
        document.forms[0].action = '?go = 提交';          //表单提交标记
        document.forms[0].submit();
}
function checkAll(touch,name)
{
    var el = document.getElementsByTagName('input'); //读取表单内所有元素
    var len = el.length;                             //读取元素的个数
    for(var i = 0; i < len; i ++ )                   //循环元素个数的次数
    {
      if((el[i].type == "checkbox") && (el[i].name == name))//读取所有名为 name 的复选框
      {
          if(document.getElementById(touch).checked)//若用户单击"全选"
          {
              el[i].checked = true;                 //选中所有复选框
          }
          else{                                     //用户取消全选
              el[i].checked = false;                //取消所有复选框的选中
          }
        }
    }
}
function chgFace(){
  document.forms[0].submit();
}
</SCRIPT>
</HEAD>
<%
uid = Trim(Request("uid"))
pwd = Trim(Request("pwd"))
sex = Trim(Request("sex"))
userFace = Trim(Request("userFace"))
email = Trim(Request("email"))
character = Trim(Request("character"))
interest = Trim(Request("interest"))
go = Trim(Request("go"))
If go <>"" Then
%>
提交成功!
<%
End If
%>
<BODY>
<CENTER>
<FORM action = "" method = "post">
<TABLE class = "table_Tb">
<CAPTION>注册页面</CAPTION>
   <TR>
      <TD class = "TdTitle">用户名</TD>
      <TD class = "TdContent">
```

```
        <INPUT TYPE="text" NAME="uid" VALUE="<% =uid%>">
      </TD>
    </TR>
    <TR>
      <TD class="TdTitle">密码</TD>
      <TD class="TdContent"><INPUT TYPE="password" NAME="pwd" VALUE="<% =pwd%>"></TD>
    </TR>
    <%
    If sex = "男" Then
      c1 = " checked"
    ElseIf sex = "女" Then
      c2 = " checked"
    End If
    %>
    <TR>
      <TD class="TdTitle">性别</TD>
      <TD class="TdContent">
        <INPUT TYPE="radio" NAME="sex" VALUE="男"<% =c1%>>男
        <INPUT TYPE="radio" NAME="sex" VALUE="女"<% =c2%>>女
      </TD>
    </TR>
    <TR>
      <TD class="TdTitle">头像</TD>
      <TD class="TdContent">
        <SELECT name="userFace" onchange="chgFace();">
          <OPTION value="">-请选择-</OPTION>
            <%
            userFace=Trim(Request("userFace"))
            For i = 1 To 60
              If userFace="Image"&i&".gif" Then
                ss=" selected"
              Else
                ss=""
              End If
            %>
          <OPTION value="Image<% =i%>.gif"<% =ss%>>Image<% =i%></OPTION>>
            <% Next%>
        </SELECT>
        <% If userFace<>"" Then Response.Write ("<IMG src=""../images/"&_
             "userface/"&userFace&""">") %>
      </TD>
    </TR>
    <TR>
      <TD class="TdTitle">电子邮件</TD>
      <TD class="TdContent"><INPUT TYPE="text" NAME="email" VALUE="<% =email%>"></TD>
    </TR>
    <%
  chrString="多重性格^乐天达观^成熟稳重^幼稚调皮^温柔体贴^活泼可爱^普普通通^"&_
            "内向害羞^外向开朗^心地善良^聪明伶俐^善解人意^风趣幽默^思想开放^"&_
            "积极进取^小心谨慎^郁郁寡欢^正义正直^悲观失意^好吃懒做^处事洒脱^"&_"
  %>
```

```
    <TR>
        <TD class="TdTitle">性格
          <input type="checkbox" name="caChr" onclick="checkAll('caChr','character')" ></TD>
        <TD class="TdContent">
          <%
          arrChrString=Split(chrString,"^")
          For i = 0 To UBound(arrChrString)
            If UBound(arrChrString)<>0 Then chrString=arrChrString(i)
           'response.write character&"."&"chrString"&"<BR>"
              If InStr(character,chrString)<>0 Then
                c3=" checked"
              Else
                c3=""
              End If
           %>
          <INPUT TYPE="checkbox" NAME="character" ID="<% =chrString %>"
                 VALUE="<% =chrString %>"<% =c3 %>>
          <LABEL for="<% =chrString %>"><% =chrString %></LABEL>
          <%
         If i Mod 5 = 4 Then Response.Write ("<BR>")
          Next
           %>
        </TD>
    </TR>
<%
insString="唱歌跳舞^体育运动^阅读书籍^舞刀弄枪^遨游网络"
%>
    <TR>
        <TD class="TdTitle">兴趣爱好
        <INPUT type="checkbox" name="caIns" onclick="checkAll('caIns','interest')"></TD>
        <TD class="TdContent">
         <%
         arrInsString=Split(insString,"^")
          For i = 0 To UBound(arrInsString)
            If UBound(arrInsString)<>0 Then insString=arrInsString(i)
              If InStr(interest,insString)<>0 Then
                c4=" checked"
              Else
                c4=""
              End If
           %>
          <INPUT TYPE="checkbox" NAME="interest" ID="<% =insString %>"
                 VALUE="<% =insString %>"<% =c4 %>>
          <LABEL for="<% =insString %>"><% =insString %></LABEL>
          <% Next %>
        </TD>
    </TR>
</TABLE><BR><BR>
<INPUT TYPE="button" NAME="go" VALUE=" 提交 " onclick="return checkForm();">
</FORM>
</BODY>
</HTML>
```

3.8 本章小结

JavaScript是一种基于对象和事件驱动的脚本语言。JavaScript代码在客户端浏览器被解释执行。它可以嵌入到HTML文件中,用户通过JavaScript输入信息时,不传给服务器,直接在本地客户端进行相应的程序处理,省去了网络传输花费的时间。

在JavaScript语言中也有固定的语法,比如变量、数据类型、表达式、选择语句、循环语句等。JavaScript是基于对象的语言,采用事件驱动的方式,使得输入简单化,直接可以操作鼠标或键盘进行操作。

3.9 思考与练习

1. JavaScript语言的特点有哪些?
2. 简要叙述JavaScript的运行原理。
3. 编写用户选择图像的功能:将图像名称设置成下拉列表控件,当用户选择某个图像后,可以在页面中将这个图像显示出来。
4. 能够利用JavaScript设计并实现网站的注册页面(参考项目案例13)。
5. 编写一个JavaScript代码设计的数字时钟。
6. 利用JavaScript编写水中倒影效果。

第二篇

ASP、JSP基础

第4章 ASP内置对象

ASP的对象之所以叫内置对象，是因为这些对象是由运行环境自动产生的，而无须用户自己去创建它们。也就是说，不需要实例化过程就能直接使用这些对象所提供的属性、方法和集合。

学习目标及要求

(1) 掌握Response对象的使用方法。

(2) 掌握Request对象的使用方法。

(3) 能够运用Application实现计数功能。

(4) 利用Session对象保存对象信息。

(5) 了解Excute、Transfer方法和Response.Redirect的区别。

4.1 ASP内置对象简介

ASP的6个内置对象是设计功能强大的Web应用程序的基础。下面对这6个对象作一简单介绍。

(1) Response：将信息发送给客户端浏览器。

(2) Request：从客户端浏览器取得信息。

(3) Session：存储一个Session会话过程中的用户信息，它仅可被该用户访问。

(4) Application：在一个ASP应用程序中存储给不同客户端共享的信息。

(5) Server：提供Web服务器的一些组件。

(6) ObjectContext：提交撤销ASP脚本初始化的事务。

4.2 Request对象

通过Request对象，可以收集到客户端浏览器的Form、Cookies、超链接和服务器的环境变量。因此，Request为ASP程序提供执行时的数据，如果没有Request对象，那么ASP程序就失去了与客户端联系的交互功能。

4.2.1 Request对象的属性与方法

Request对象提供了4个集合，可以用来访问客户端对Web服务器请求的各类信息，这

些集合如表 4-1 所示。

表 4-1 Request 对象的集合及说明

集合名称	说　明
ClientCertificate	当客户端访问一个页面或其他资源时,用来向服务器表明身份的客户证书的所有字段或条目的数值集合,每个成员均为只读
Form	METHOD 的属性值为 POST 时,所有作为请求提交的<FORM>段中的 HTML 控件单元的值的集合,每个成员均为只读
QueryString	依附于用户请求的 URL 后面的名称/数值对或者作为请求提交的且 METHOD 属性为 GET(或者省略其属性)的,或<FORM>中所有 HTML 控件单元的值,每个成员均为只读
ServerVariables	随同客户端请求发出的 HTTP 报头值,每个成员均为只读

Request 对象唯一的属性及说明如表 4-2 所示,它提供用户请求的字节数量的信息,它很少用于 ASP 页,我们通常关注指定值而不是整个请求字符串。

表 4-2 Request 对象的属性

属性	说　明
TotlBytes	只读,返回由客户端发出的请求的整个字节数量

Request 对象唯一的方法及说明如表 4-3 所示,它允许访问从一个<FORM>段中传递给服务器的用户请求部分的完整内容。

表 4-3 Request 对象的方法及说明

方法	说　明
BinaryRead(count)	当数据作为 POST 请求的一部分发往服务器时,从客户请求中获得 Count 字节的数据,返回一个 Variant 数组(或者 SafeArray)。如果 ASP 代码已经引用了 Request.Form 集合,这个方法就不能用。同时,如果用了 BinaryRead 方法,就不能访问 Request.Form 集合

4.2.2 Request.Form 集合

Form 数据集合是 Request 对象中最常用的数据集合,它检索的是以 POST 方式提交的 HTML 表单元素的数据。其语法如下:

```
表单对象内容 = Request.Form("表单对象名称")或
表单对象内容 = Request.Form("索引值")
```

其中,“表单对象名称”是要检索的表格元素的名称,“索引值”是访问某参数中多个值中的一个。

Request 是 ASP 对象,Form 则是 Request 对象所包含的对象集合。

【例 4-1】 用户名和密码的提交。

【代码】

```
<FORM action = "5 - 4b.asp" method = "post">
  用户名: < INPUT type = "text"? name = "yourname">< BR >
```

```
  密码：<INPUT type="password" name="yourpwd"><BR>
  <INPUT type="submit" value="submit">
</FORM>
<% Dim jvName,jvPass
  jvName =Request.Form("jvName")
  jvPass=Request.Form("jvPass")
  Response.Write "您的用户名"
  Response.Write jvName
  Response.Write "和密码是"
  Response.Write jvPass %>
```

4.2.3 Request.QueryString 集合

QueryString 集合用来收集以 GET 方式提交的数据表单，而以 POST 方式提交的表单数据则用 Form 集合来收集。

QueryString 数据集合与 Form 数据集合在实现上没有太大的区别，主要的区别在于前者是通过取得 HTTP 的附加参数来传递数据(附加参数通常是使用“?”来连接的，多个参数间用“&”隔开)，而后者是通过单独的数据包来传递数据。

【例 4-2】 输入信息平台表单。

【代码】

```
<FORM action="5-5b.asp" method="get">
      your name:<INPUT tpye="text"? name="yourname"><BR>
      your pwd: <INPUT type="password" name="yourpwd"><BR>
      <input type="submit" value="submit">
    </FORM>
<% Dim jvName,jvPass
  jvName =Request.Querystring("jvName")
  jvPass =Request.Querystring("jvPass")
  Response.Write "您的用户名"
  Response.Write jvName
  Response.Write "和密码是"
  Response.Write jvPass %>
```

比较：Request.Form 和 Request.QueryString 的区别

Request.Form 提取用 Post 方法发送的表单数据；而 Request.Querystring 提取 URL 参数传递的值或以 Get 方法提交的表单数据，比如常用的“隐藏域”。

4.2.4 Request.ServerVariables 集合

ServerVariables 数据集合负责取得 Web 服务器端的环境变量(server environment variable)，检索预定的环境变量。

同样，可以利用环境变量的名称来取得指定的环境变量的值，语法如下：

```
Request.ServerVariables("环境变量的名称")
```

其中,“环境变量的名称”表示环境变量名称字符串,常用的环境变量如下。

- LOCAL_ADDR:服务器端 IP 地址。
- REMOTE_ADDR:客户端 IP 地址。
- SCRIPT_NAME:正在运行的脚本文件的路径信息。

【例 4-3】 显示来访者的 IP 地址(如图 4-1 所示)。

```
<%
Response.Write  "来访者 IP 地址是: "
Response.Write
Request.ServerVariables
("Remote_Addr")
%>
```



图 4-1 显示来访者的 IP 地址

总结:Request 对象的使用方法。

Request 对象包含 3 个对象:form、querystring、servervarivables。ASP 内置对象除了对象集合外还有对象属性、对象方法。Request 对象的对象属性只有一个,就是 TotalBytes(接受的字节数),可将<%=request.totalbytes%>语句加入到任何一个接收数据的 ASP 页面中进行显示。Request 对象的对象方法只有一个:BinaryRead,此属性不常用。

4.3 Response 对象

Response 对象可以把 ASP 程序在服务器执行后的结果发送到客户端的浏览器,让客户可以看见结果,如图 4-2 所示。同时,也可以通过 Response 对象设置客户端浏览器的状态,比如设置 Cookies,Cookies 是一种跟踪客户端状态的技术,网站利用 Cookies 来记录访问网站的次数频率,而 Cookies 就是那些保存在 Windows 系统目录 Cookies 里的文件,这些文件可以供网站重新读取和设置,用来跟踪客户的状态。Request 和 Response 两者功能紧密配合,一个负责收集,另一个负责输出。

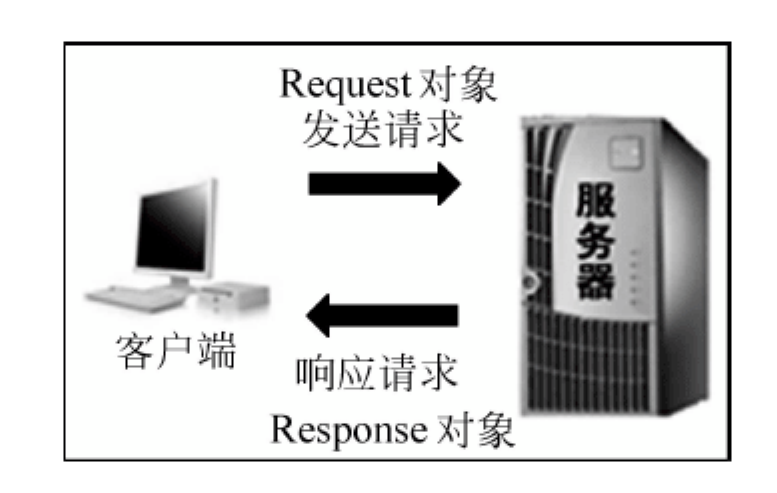


图 4-2 Request 和 Response 客户端和服务器示意图

Response 对象的主要功能是由服务端发送信息到客户端。Response 对象属性和方法如表 4-4 所示。

表 4-4 Response 对象属性和方法

	名 称	含 义	举 例
方法	Response. Write()	向客户端浏览器发送信息	Response. Write("Hello World!")
	Response. Redirect()	重定向客户端浏览器	Response. Redirect ("../Text. asp")
	Response. Cookies()	写入客户端 Cookie 值	Response. Cookies("Login")="True"
	Response. Flush()	清除缓冲	Response. Flush()
	Response. End()	结束 ASP 程序的执行	Response. End()
属性	Response. Buffer	缓冲页面	Response. Buffer=True/False

4.3.1 Response 对象的属性和方法

Response 对象只有一个集合，如表 4-5 所示，该集合设置希望放置在客户系统上的 Cookie 的值，它直接等同于 Request. Cookies 集合。

表 4-5 Response 对象集合

集合名称	说 明
Cookies	在当前响应中发回客户端的所有 Cookie 的值，这个集合为只写

Response 对象也提供一系列的属性，可以读取(多数情况下)和修改，使响应能够适应请求。这些由服务器设置，我们无须设置它们。需要注意的是，当设置某些属性时，使用的语法可能与通常所使用的有一定的差异。Response 对象属性与说明如表 4-6 所示。

表 4-6 Response 对象属性与说明

属 性	说 明
Buffer=True|False	读/写，布尔型。表明由一个 ASP 页所创建的输出是否一直存放在 IIS 缓冲区，直到当前页面的所有服务器脚本处理完毕或 Flush、End 方法被调用。在任何输出(包括 HTTP 报送信息)送往 IIS 之前这个属性必须设置。因此在. asp 文件中，这个设置应该在<%@LANGUAGE=…%>语句后面的第一行。ASP 3.0 默认设置缓冲为开(True)，而在早期版本中默认为关(False)
CacheControl“setting”	读/写，字符型。设置这个属性为“Public”允许代理服务器缓存页面，如为“Private”则禁止代理服务器缓存的发生
Charset=“value”	读/写，字符型。在由服务器为每个响应创建的 HTTP Content-Type 报头中附上所用的字符集名称(例如 ISO-LATIN-7)
Content Type=“MIME-type”	读/写，字符型。指明响应的 HTTP 内容类型，标准的 MIME 类型(例如“text/xml”或者“Image/gif”)。假如缺省，表示使用 MIME 类型“text/html”，内容类型告诉浏览器所期望内容的类型
Expires minutes	读/写，数值型。指明页面有效的以分钟计算的时间长度，假如用户请求其有效期满之前的相同页面，将直接读取显示缓冲中的内容，这个有效期过后，页面将不再保留在私有(用户)或公用(代理服务器)缓冲中
Expires Absolute #date [time]#	读/写，日期/时间型。指明当一个页面过期和不再有效时的绝对日期和时间

续表

属 性	说 明
IsClientConnected	只读，布尔型，返回客户是否仍然连接和下载页面的状态标志。在当前的页面已执行完毕之前，假如一个客户转移到另一个页面，这个标志可用来中止处理（使用 Response. End 方法）
PICS（"PICS-Label-string"）	只写，字符型。创建一个 PICS 报头定义页面内容中的词汇等级，如暴力、性、不良语言等
Status＝"Code message"	读/写，字符型。指明发回客户的响应的 HTTP 报头中表明错误或页面处理是否成功的状态值和信息。例如"200 OK"和"404 Not Found"

Response 对象提供一系列的方法，如表 4-7 所示，允许直接处理为返回给客户端而创建的页面内容。

表 4-7 Response 对象的方法与说明

方 法	说 明
AddHeader（"name"，"content"）	通过使用 name 和 Content 值创建一个定制的 HTTP 报头，并增加到响应之中，不能替换现有的相同名称的报头，一旦已经增加了一个报头就不能被删除。这个方法必须在任何页面内容（即 text 和 HTML）被发往客户端前使用
AppendToLog（"string"）	当使用 W3C Extended Log File Format 文件格式时，对于用户请求的 Web 服务器的日志文件增加一个条目。至少要求在包含页面的站点的 Extended Properties 页中选择 URL Stem
BinaryWrite（safeArray）	在当前的 HTTP 输出流中写入 Variant 类型的 SafeArray，而不经过任何字符转换。对于写入非字符串的信息，例如定制的应用程序请求的二进制数据或组成图像文件的二进制字节是非常有用的
Clear（）	当 Response. Buffer 为 True 时，从 IIS 响应缓冲中删除现存的缓冲页面内容，但不删除 HTTP 响应的报头，可用来放弃部分完成的页面
End（）	让 ASP 结束处理页面的脚本，并返回当前已创建的内容，然后放弃页面的任何进一步处理
Flush（）	发送 IIS 缓冲中所有当前缓冲页给客户端。当 Response. Buffer 为 True 时，可以用来发送较大页面的部分内容给个别的用户
Redirect（"url"）	通过在响应中发送一个"302 Object Moved"HTTP 报头，指示浏览器根据字符串 url 下载相应地址的页面
Write（"string"）	在当前的 HTTP 响应信息流和 IIS 缓冲区写入指定的字符，使之成为返回页面的一部分

4.3.2 Response. Write 方法

该方法把数据以 HTML 格式发送到客户端浏览器，向客户端输出信息。

语法：

```
Response.Write 变量数据或字符串
如：Response.Write "您来访的日期是" & Date()          '输出一个字符串表达式'
```

简略语法：

```
<% = 变量数据或字符串 %>
```

【例 4-4】 向屏幕输出“Hello，world！”。

```
<% Response.write "Hello,world!" %>
```

这样，会在客户端浏览器上左端顶角显示“Hello，world！”字样。当然也可以用下段程序代替：

```
<% = "Hello,world!" %> ze = n face = charnames color = ＃n>
```

4.3.3 Response.Redirect 方法

Redirect 方法实现页面重定位，使浏览器可以重新定位到另一个 URL 上。这样，当客户发出 Web 请求时，客户端的浏览器类型已经确定，客户被重新定位到相应的页面。如下面两个程序是根据在表单中不同的选择，而重新定位到不同的 URL 页面上。

语法为：

```
Response.Redirect url
```

其中，url 表示相对路径、绝对路径或 URL 字符串。

例如：

```
<%
    Response.Redirect "http://www.sina.com.cn"        '引导至新浪网
    Response.Redirect "4 - 1.asp"                      '引导至站内其他网页
    Response.Redirect strURL                           '引导至变量表示的网址
%>
```

【例 4-5】 Redirect 方法的使用。

程序 1：表单示例(如图 4-3 所示)。

```
<HTML>
<HEAD>
<TITLE>5 - 2.asp 示例</TITLE>
</HEAD>
<BODY>
<FORM aciton = "5 - 2b.asp" method = "post">
<SELECT name = "wheretogo">
<OPTION selected value = "163">163 </OPTION>
<OPTION value = "ccniit"> ccniit </OPTION>
</SELECT>
<INPUT type = submit name = "jump" value = "Jump">
</FORM>
</BODY>
</HTML>
```

程序 2：表单处理及转向程序(如图 4-4 所示)。

```
<% response.buff = true %>
<HTML>
```

```
<HEAD>
<TITLE>5-2b.asp Redirect 示例</TITLE>
</HEAD>
<BODY>
<%
where = Request.form("wheretogo")
Select Case where
 case "163"
response.redirect " http://www.163.com/"
 case "ccniit"
  response.redirect " http://www.ccniit.com/"
End Select
 %>
</BODY>
<HTML>
```



图 4-3 表单示例



图 4-4 表单处理及转向程序

4.3.4 Response.Cookies 集合

Response 对象的 Cookies 数据集合是负责取得记录在客户端的 Cookies 数据，允许用户检索在 HTTP 请求中发送的 Cookie 的值。其语法如下：

```
CookieValue = Request.Cookies(CookieName)[(key)|.Attribute]
```

其中，Key 参数表示可以从 Cookies 目录中取得 Cookies 的值，即用于从 Cookie 字典中检索关键字的值，是一个选择性参数。

获取客户端 Cookies 信息，前提是已经使用 Response. Cookies（“变量”）写入 Cookies 中数据。

设置 Cookies 的方法：

1）设置不含关键字的单值 Cookie

设置单值 Cookie 很简单，只要指定 Cookie 名称和它的值即可。例如：

```
<% Response.Cookies("strName") = "高航"   %>
```

2）设置含关键字的多值 Cookie

多值 Cookie 就类似于一个数组，可以包含多个元素，分别用关键字指定即可。例如：

```
<%
Response.Cookies("strUser")("name") = "刘晶"
Response.Cookies("strUser")("age") = 23
%>
```

3）设置 Cookie 的有效期

如果不设置 Cookie 的有效期，则关闭浏览器后该 Cookie 就消失了，下面就针对上面的例子设置有效期：

```
<%
Response.Cookies("strName").Expires = #2012-1-8#          '设置单值 Cookie 有效期
Response.Cookies("strUser").Expires = #2012-1-8#          '设置多值 Cookie 有效期
%>
```

【例 4-6】 Cookies 集合的用法。

```
<% Response.Cookies("Login") = "true"
    If Request.Cookies("Login") = "true" Then
        Response.Write ("欢迎登录!")
    Else
        Response.Write ("请先登录!")
End If %>
```

小知识：个人计算机的 Cookie 设置

由于 IE 浏览器的默认设置是“中级”，即限制部分网站对 Cookie 的利用，所以要进行 Cookie 的设置。

操作方法：在 IE 中选择菜单“工具”→“Internet 选项”→“隐私”查看和修改 Cookie。

4.3.5　Response.End 方法

End 方法用于告知 Active Server 当遇到 End 时停止处理 ASP 文件。如果 Response 对象的 Buffer 属性设置为 True，这时 End 方法即把缓存中的内容发送到客户并清除缓冲区。所以要取消所有向客户的输出，可以先清除缓冲区，然后利用 End 方法，如下面程序所示，程序运行结果如图 4-5 所示。

```
<%   response.Write "hello"
response.End
response.Write " world"   %>
```

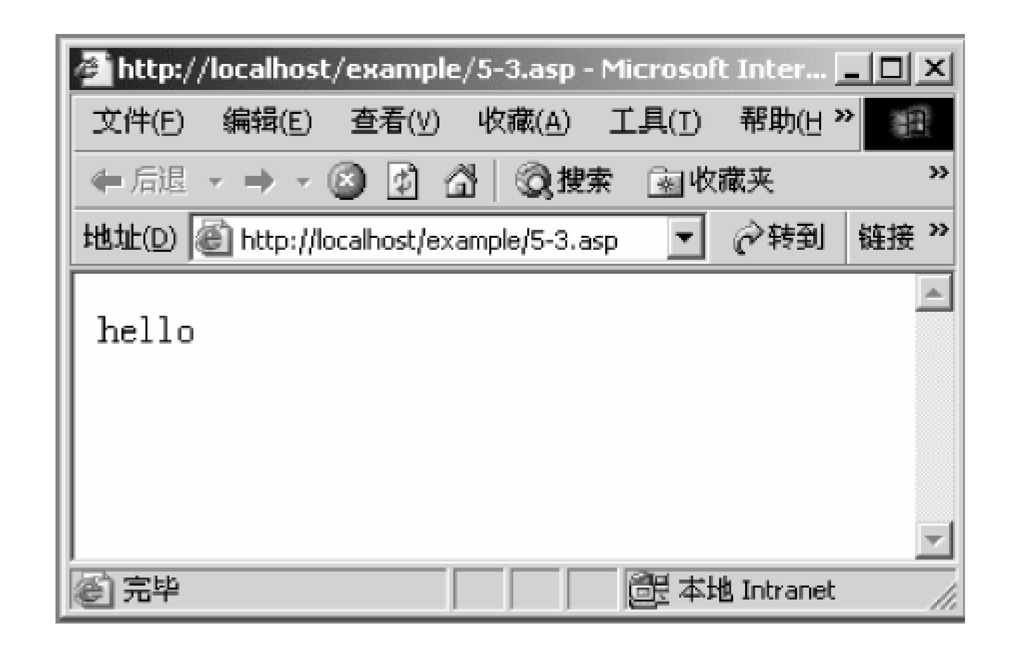


图 4-5 End 方法示意图

4.4 Application 对象

4.4.1 Application 对象的属性和方法

Application 对象是个应用程序级的对象,用来在所有用户间共享信息,并可以在 Web 应用程序运行期间持久地保持数据。Application 对象成员包括 Application 对象的集合、方法和事件,如表 4-8 所示。

表 4-8 Application 对象的方法与说明

集合和方法	描 述
Contents	没有使用<object>元素定义的存储于 Application 对象中的所有变量和其值的集合,包括 Variant 数组和 Variant 类型对象实例的引用
StaticObjects	使用<object>元素定义的存储于 Application 对象中的所有变量和其值的集合
Contents. Remove("variable_name")	从 Application. Contents 集合中删除名为 variable_name 的变量
Contents. RemoveAll()	从 Application. Contents 集合中删除所有变量
Lock()方法	锁定 Application 对象,使得只有当前的 ASP 页面可访问数据。用于确保允许通过两个用户同时读取和修改该值的方法而进行的并发操作不会破坏数据
Unlock()方法	解除在 Application 对象上的 ASP 网页锁定
OnStart 事件	在建立 Application 对象时产生该事件,用于初始化变量、创建对象和运行其他代码
OnEnd 事件	在结束 Application 对象时产生该事件,释放存储在应用程序中的所有变量

Application 对象的作用是记载所有用户的共享信息。

Application 对象的有效期是从应用程序启动后第一个用户开始访问到所有用户都结束访问。

Application 对象的主要数据集合是 Contents,其作用是获取脚本中添加的 Application

变量集合。

Application 对象的方法如下。

- Lock：锁定 Application 对象，此时其他用户就不能再修改 Application 中变量的值。
- UnLock：解除锁定的 Application 对象。

Application 对象的事件如下。

- Application_OnStart：当应用程序的第一个用户访问时，会触发该事件，执行其中的语句。
- Application_OnEnd：当应用程序关闭时触发该事件，执行其中的语句。

【例 4-7】 Application 对象的应用。

```
<% Dim itemName, itemPwd, itemTime
  Application ("itemName") = "启蒙星 2.0"
  Application ("itemPwd")  = "62768866"
  Application ("itemTime")  = #2008 - 9 - 1#
  itemName  = Application ("itemName")
  itemPwd = Application ("itemPwd")
  itemTime = Application ("itemTime")
  Response.Write Application("itemName")&"在线学习系统的通用密码为"&Application("itemPwd")&",
发布时间为"&Application("itemTime") %>
```

【例 4-8】 Application 对象实现的计数器。

```
<HTML>
<BODY>
    <H2 align = "center">my web</H2>
    <%
        Application.Lock      '锁定
        Application("INTall") = Application("INTall") + 1   '计数
        Application.Lock      '解除锁定
            Dim visit
            visit = Application("INTall")      '获取变量值
        Response.Write"<p>您是第"&visit&"位访客。"
     %>
</BODY></HTML>
```

4.4.2 使用 Application 对象制作网络聊天室

网络聊天室是一个比较复杂的系统。由于要在多个用户之间显示聊天内容，会用到 Application 对象，在传递用户的信息时，可以使用 Session 对象。此外，还将频繁使用 Request 和 Response 等对象。聊天室中需要使用 ASP 的多个对象，并根据实际情况设计出符合要求的程序。需要用到的 Application 的操作存储信息的语法如下：

```
Application(variable_name) = 变量或字符串信息
```

保存 Application 信息时，需要先锁定，然后再解除锁定。

利用 Application 对象存储数组信息和 Session 对象类似，也必须把数组当成一个整体存入和读取，只是存储时别忘了 Lock 和 Unlock 就行了。语法如下：

```
Application("Application 数组") = 数组变量
```

获取 Application 信息的语法如下：

```
a = Application("Session 数组")
```

【例 4-9】 存储示例。

```
<%
    Dim strNames(1)                              '声明一个长度为 2 的数组
    strNames(0) = "白芸"
    strNames(1) = "海霞"
    Application.Lock                             '锁定
    Application("strNames") = strNames           '给 Application 变量赋值
    Application.Unlock                           '解除锁定
%>
```

下面是读取示例：

```
<%
    Dim strNames                                 '声明一个普通变量
    strNames = Application("strNames")           '读取 Application 变量
    Response.Write strNames(0) &                 "您好,欢迎您<br>"
    Response.Write strNames(1) &                 "您好,欢迎您<br>"
%>
```

项目案例 1：网络聊天室(如图 4-6 所示)。

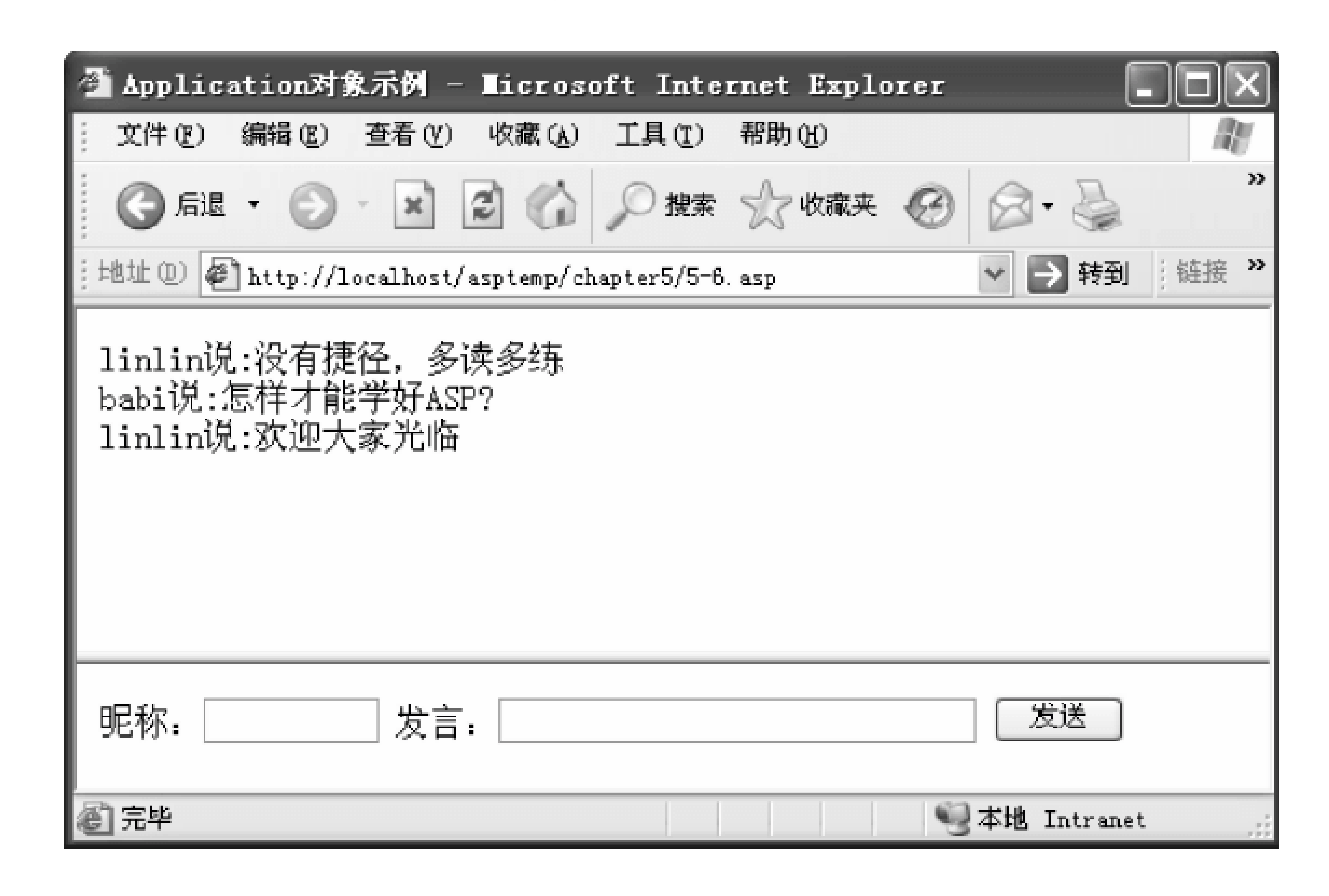


图 4-6 网络聊天室示意图

【代码】

1) frame.asp

```
<HTML>
<FRAMESET rows = " * ,60">
        <FRAME name = "message" src = " chart1.asp">
        <FRAME name = "say" src = "chart2.asp">
</FRAMESET>
```

```
</HTML>
```

2）chart1.asp

```
<HTML>
<HEAD>
        <TITLE>留言页面</TITLE>
        <META http-equiv="refresh" content="5">
</HEAD>
<BODY>
      <% Response.Write Application("strChat") %> '显示聊天的内容
</BODY>
</HTML>
```

3）chart2.asp

```
<HTML>
<BODY>
        <FORM name="form1" method="post" action="">
        昵称：<INPUT type="text" name="txtName" size="10">
        发言：<INPUT type="text" name="txtSay" size="30">
        <INPUT type="submit" value="发送">
          </FORM>
          <%
          '如果提交了表单，将发言内容添加到 application 对象中
        if Trim(Request.Form("txtName"))<>""And
        Trim(Request.Form("txtSay"))<>""Then
        '获取本次发言字符串，包括发言人和发言内容
        Dim strSay
        strSay=Request.Form("txtName")&"说："&Request.Form("txtSay")&"<br>"
        '将本次发言添加到聊天内容中
        Application.Lock              '锁定
        Application("strChat") = strSay & Application("strChat")
        Application.Unlock            '解锁
        End if
          %>
</BODY>
</HTML>
```

4.5 Session 对象

HTTP 协议是一种无状态(stateless)的协议，利用 HTTP 协议无法跟踪用户。服务器完成用户的请求后，服务器将不能再继续保持与该用户浏览器的连接。利用 Session 可以存储浏览者的一些特定信息。

在 ASP 中 Session 代表了服务器与客户端之间的“会话”。

Session 信息保存在服务器端的一块内存区域中。当每一个用户开始访问时(也称开始一个会话时)，服务器端就会给该用户建立一个 Session 对象，在服务器端内存中给他分配一块区域，用来存储该用户的信息。当该用户结束访问时(也称结束一个会话时)，服务器端

就会清除这个 Session 对象，并释放相应的内存区域。

Session 对象的主要数据集合如下。

- Contents：获取脚本中添加的 Session 变量集合。
- Session：对象的主要属性。
- SessionID：返回会话标识符(ID)，即存储用户的 Session ID。
- TimeOut：定义 Session 的有效期。单位为分钟，默认为 20 分钟。
- Abandon：清除当前的 Session 对象。

Session 对象包括以下两个事件。

- Session_OnStart：当开始一个会话时，触发该事件。
- Session_OnEnd：当结束一个会话时，触发该事件。

不过这两个事件只能用在 Global.asa 文件中。

利用 Session 可以保存变量或字符串等信息，语法如下：

```
Session("Session 名字") = 变量或字符串信息
```

保存 Session 信息的例子：

```
Session("strName") = "卓云"
Session("strA") = strA
```

读取 Session 信息的例子：

```
strName = Session("strName ")
strB = Session(" strA ")
```

比较：Application 和 Session 的区别。

Application 对象和 Session 对象有很多相似之处，它们的功能都是用于在不同的 ASP 之间共享信息。

最大的不同在于其应用的范围：Application 对象是所有用户，Session 对象是单一用户。

ASP 的 Application 和 Session 对象体现了其他 ASP 内置对象所没有的特征事件。每一个访客访问服务器时都会触发一个 OnStart 事件(第一个访客会同时触发 Application 和 Session 的 OnStart 事件，但 Application 先于 Session)，每个访客的会话结束时都会触发一个 OnEnd 事件(最后一个访客会话结束时会同时触发 Application 和 Session 的 OnEnd 事件，但 Session 先于 Application)。

【例 4-10】 Session 对象的应用。

Session 对象是针对每个浏览网页的用户而言的。浏览者浏览网页时，网站分配给浏览者不相同的 Session，用于存放用户的身份等信息。

```
<% If jvName = "QMX" And jvPwd = "62768866" Then
       Response.Redirect ("/main.asp")
       Session ("Login")  = true
  Else
       Response.Redirect ("/login.asp")
  End If %>
<%
  Dim pageView
```

```
  If Session ("pageView") = "" Then
    Session ("pageView") = 1
  Else
    Session ("pageView") = Session ("pageView") + 1
  End If
  Response.Write "您是第"&Session ("pageView")&"次光临本页!"
%>
```

【例 4-11】 Session 对象的属性 TimeOut 的应用。

```
<%
    Response.Write ("您的 Session 的 ID 号是:")
    Response.Write Session.SessionID&"<BR>"
    Response.Write "Session 默认的过期时间是:"
    Response.Write Session.TimeOut&"分钟,<BR>"
    Session.TimeOut = 30
    Response.Write ("您的 Session 被重置,重置后的过期时间为:")
    Response.Write Session.TimeOut&"分钟。"
%>
```

小知识：Global.asa 文件

Global.asa 文件是手动编写的文件。操作方法：新建 TXT 文件,将后缀改为.asa,放置于相应网站的根目录。Global.asa 文件执行优先于任何网页文件。

【例 4-12】 在线人数统计。

```
<SCRIPT Language = VBScript RUNAT = Server>
   Sub Application_OnStart '初始值为 0
   Application ("OnLine") = 0
   End Sub
   Sub Session_OnStart '一个用户访问进行记数加 1
   Application.Lock
   Application ("OnLine") = Application ("OnLine") + 1
   Application.Unlock
   End Sub
   Sub Session_OnEnd '一个用户关闭网页,记数减 1
   Application.Lock
   Application ("OnLine") = Application ("OnLine") - 1
   Application.Unlock
   End Sub
</SCRIPT>
使用代码:
...
当前在线人数<% = Application("Online") %>
...
```

4.6 Server 对象

Server 对象提供对服务器上的方法和属性的访问,其中大多数方法和属性是为程序的功能服务的。有了 Server 对象,用户就可以在服务器上启动 ActiveX 对象例程,并使用

Active Server 服务提供像 HTML 和 URL 编码这样的函数。

Server 对象的使用语法为：

```
Server.property|method
```

其中，property 表示 Server 对象的属性，method 表示 Server 对象的方法。Server 对象的方法如表 4-9 所示。

表 4-9 Server 对象的方法

方 法	描 述
CreateObject	创建服务器组件的实例
HTMLEncode	将 HTML 编码应用到指定的字符串
MapPath	将指定的虚拟路径，无论是当前服务器上的绝对路径，还是当前页的相对路径，映射为物理路径
URLEncode	将 URL 编码规则，包括转义字符，应用到字符串
Transfer	直接输出.asp 文件请求，不必离开服务器队列
Execute	执行另一个.asp 文件，然后返回当前文件
GetLastError	获取 ASP 的错误

Server 的属性只有 ScriptTimeout，表示超时值，即在脚本运行超过这一时间之后作超时处理。系统的默认值为 90 秒。

项目案例 2：购物车案例。

【题目】 购物车的主要功能包括向购物车中添加商品、查看购物车、修改已购买商品的数量、移除购物车中指定商品，如图 4-7 所示。



图 4-7 购物车示意图

1）需求描述

用户在商品详细信息页单击“购买”后，进入购物车页面，该页面使用一个表格存放用户选择过的所有商品信息。

2）操作技能

- 使用 Cookie 保存用户信息。
- 使用字符串拼接及拆分的方法添加新记录。

- 使用子过程处理通用问题。
- 读取 Cookie 中保存的信息。

3）实现思路

单击“购买”某商品后，进入购物车页面。首先读取 Cookie 中的字符串，在 Cookie 字符串中查找新提交的商品是否已存在，通过对变量 position 的判断确定该商品是否已被购买过，若变量的值不为 0，则只修改购买数量，不添加新的记录，将 Cookie 中的商品信息显示在购物车页面中。

【代码】　cart.asp

```
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=gb2312" />
<TITLE>购物车</TITLE>
<LINK href="css/public.css" rel="stylesheet" type="text/css" />
</HEAD>
<BODY>
<TABLE width="230" border="0" cellspacing="0" cellpadding="0" align="center">
    <TR>
      <TD><IMG src="images/detail01.jpg" border="0" usemap="#Map" />
      <MAP name="Map" id="Map">
        <area shape="rect" coords="548,10,605,27" href="cart.asp" target="_blank" />
    </MAP></TD>
</TR>
</TABLE>


<TABLE width="830" border="0" align="center" cellpadding="0" cellspacing="0" style=
"border:solid 1px #D8EBEB;">
<TR>
  <td height="23" align="left" bgcolor="#F0F0F0" class="d_h2"> 我的购物车</TD>
</TR>
<TR>
  <TD height="330" align="center"><%
proId=Trim(Request("proId"))
proName=Trim(Request("proName"))
'response.write proId&"<hr size=1 color=red>"
price=Trim(Request("price"))
number=Trim(Request("number"))
go=Trim(Request("go"))
command=Trim(Request("command"))
'Response.Cookies("PorId")=""
'Response.Cookies("proNameString")=""
'Response.Cookies("priceString")=""
'Response.Cookies("numberString")=""
proIdString=Request.Cookies("proIdString")
proNameString=Request.Cookies("proNameString")
priceString=Request.Cookies("priceString")
numberString=Request.Cookies("numberString")
'response.write proIdString&"<BR>"
```

```
'Response.End
position = InStr(proIdString&"^","^"&proId&"^")
'response.write "proIdString = "&proIdString&"<HR>proId = "&proId&"<HR color = red>
"&position&"<HR color = #6600FF>"
Sub setCookie(ckItem,ckValue)
    'response.write ckItem&"<BR>"
    'response.write ckValue&"<HR color = blue>"
    Response.Cookies(ckItem) = ckValue
    'Response.Cookies.Expires(ckItem) =
  date + 2
End Sub
If command = "del" Then
     If position <> 0 Then
      arrPI = Split(proIdString,"^")
      arrPN = Split(proNameString,"^")
      arrPR = Split(priceString,"^")
      arrNM = Split(numberString,"^")
      proIdString = ""
      proNameString = ""
      priceString = ""
      numberString = ""
      'response.write UBound(arrPN)&"<BR>"&position&"<HR
size = 1 color = blue>"
  For i = 1 To UBound(arrPN)
    If i <> (position + 1)/2 Then
      proIdString = proIdString&"^"&arrPI(i)
      proNameString = proNameString&"^"&arrPN(i)
      priceString = priceString&"^"&arrPR(i)
      numberString = numberString&"^"&arrNM(i)
    End If
  Next
  'response.write proIdString & "<HR>" & position
  'response.end
  Call setCookie("proIdString",proIdString)
  Call setCookie("proNameString",proNameString)
  Call setCookie("priceString",priceString)
  Call setCookie("numberString",numberString)
End If
End If
If go <>"" Then
If position <> 0 Then
  'response.write position&"<BR>"
  'Response.Write position&"<HR>"
  arrNM = Split(numberString,"^")
  numberString = ""
  For i = 1 To UBound(arrNM)
    If i = (position + 1)/2 Then
      newNumber = CInt(arrNM(i)) + CInt(number)
    Else
      newNumber = arrNM(i)
    End If
```

```
      numberString = numberString&"^"&newNumber
    Next
    'response.write numberString
    'response.End
      Else
        proIdString = proIdString&"^"&proId
proNameString = proNameString&"^"&proName
        priceString = priceString&"^"&price
       numberString = numberString&"^"
&number
      End If
      Call  setCookie ("proIdString", proIdString)
      Call setCookie("proNameString",proNameStri ng)
      Call  setCookie("priceString", priceString)
      Call setCookie("numberString",numberString)
ElseIf proName <>"" Then
%>
<FORM action = "" method = "post">
<TABLE width = "80%" border = "0" align = "center" cellpadding = "5" cellspacing = "1" class =
"cart">
   <TR>
      <TH width = "25%" align = "center"><STRONG>商品名称</STRONG></TH>
        <TD width = "75%" bgcolor = "#FFFFFF"><% = proName %></TD>
      </TR>
      <TR>
        <TH align = "center"><STRONG>单价</STRONG></TH>
        <TD bgcolor = "#FFFFFF"><% = price %></TD>
      </TR>
      <TR>
        <TH align = "center"><STRONG>数量</STRONG></TH>
        <TD bgcolor = "#FFFFFF"><INPUT NAME = "number" TYPE = "text" class = "input_text" value =
"1"></TD>
      </TR>
   </TABLE>
<BR><input type = "submit" name = "go" value = "加入购物车">
<INPUT TYPE = "hidden" NAME = "proName" value = "<% = proName %>">
<INPUT TYPE = "hidden" NAME = "price" value = "<% = price %>">
<INPUT TYPE = "hidden" NAME = "proId" value = "<% = proId %>">
</FORM>
<%
End If
%>
<TABLE width = "80%" border = "0" cellpadding = "5"  cellspacing = "1"  class = "cart">
<TR>
  <TH align = "center"><STRONG>编号</STRONG></TH>
  <TH align = "center"><STRONG>商品名称</STRONG></TH>
  <TH align = "center"><STRONG>单价</STRONG></TH>
  <TH align = "center"><STRONG>数量</STRONG></TH>
  <TH align = "center"><STRONG>小计</STRONG></TH>
  <TH align = "center"><STRONG>删除</STRONG></TH>
</TR>
```

```
<%
arrPI = Split(proIdString,"^")
arrPN = Split(proNameString,"^")
arrPR = Split(priceString,"^")
arrNM = Split(numberString,"^")
sum = 0
For i  = 1 To UBound(arrPN)
%>
<TR>
  <TD align = "center"><% = i %></TD>
  <TD align = "center"><a href = "detail.asp?proId = <% = arrPi(i) %> &proName = <% = arrPN
(i) %> &price = <% = arrPR(i) %>" target = "_blank"><% = arrPN(i) %></TD>
  <TD align = "center"><% = FormatNumber(arrP
R(i),2) %></TD>
      <TD align = "center"><% = arrNM(i) %></TD>
      <TD align = "center"><% = FormatNumber(arrPR(i) * arrNM(i),2) %></TD>
  <TD align = "center" style = "cursor:hand;" onclick = "if (confirm('确定要从购物车中删除此
商品?')) location.replace('cart.asp?command = del&proId = <% = arrPI(i) %>')"><IMG src =
"images/cut.gif" alt = "删除" width = "19" height = "21" /></TD>
    </TR>
<%
sum = sum + arrPR(i) * arrNM(i)
Next
%>
    <TR>
     <TD colspan = "4" align = "center">合计</TD>
  <TD  colspan = "2" align = "center"><% = FormatNumber(sum,2) %></TD>
  </TR>
</TABLE>
<BR><BR>
<INPUT type = "button" value = "去收银台结算" onclick = "location.replace('account.asp')">
<INPUT type = "button" value = "关闭购物车" onclick = "window.close();"></TD>
</TR>
</TABLE>
<TABLE width = "330" border = "0" align = "center" cellpadding = "0" cellspacing = "0">
<TR>
  <TD><IMG src = "images/index02.jpg" width = "1002" height = "367"></TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

4.7 本章小结

本章介绍了 ASP 内置对象，是整个 ASP 的基础。学习本章内容要求掌握如何利用 Request 对象 Form、QueryString、Cookies 集合获取客户端信息，利用 Response 对象的 Write 方法向客户端输出信息，利用 Application 和 Session 进行计数；会使用 Server 对象设置在 MapPath 中的相对路径与绝对路径。

通过本章学习后，能够独自开发出动态页面，比如购物车、留言板等，并在此功能上进行扩展。

4.8　思考与练习

1. 如何使用 Response 输出信息？
2. 如何使用 Request 获取信息？
3. 如何使用表单及表单元素？
4. 如何使用 Application 对象？
5. 如何使用 Session 对象？
6. 简述 Application 和 Session 的应用场合。
7. 如何向客户端写入 Cookie 数据？
8. 如何读取 Cookie 中的数据？
9. 实现网站计数功能。
10. 实现购物车功能。

第5章 ASP内置组件

在 ASP 中内置了许多功能强大的 ActiveX 服务器组件，利用这些组件就可以方便地实现数据存取、文件存取等功能。

学习目标及要求

(1) 掌握 Ad Rotator 组件的用法。

(2) 掌握 Content Linking 组件的用法。

(3) 掌握计数器组件的用法。

(4) 理解 Browser Capabilities 组件的用法。

5.1 ASP 内置组件简介

ASP 组件均遵循 Microsoft 的 ActiveX 标准。ActiveX 组件是一个文件，该文件包含执行一项或一组任务的代码，组件可以执行某些特定的功能。

安装好 ASP 平台后，所有的 ASP 内置组件都被安装注册到服务器上。调用内置组件的步骤如下：首先，必须创建组件的一个实例，并将该实例赋予一个变量；然后，调用组件的方法和属性，完成相应的功能。ASP 的内置组件有 13 种，而且还可以安装其他第三方的组件。在此只介绍几种重要的组件，如表 5-1 所示。

表 5-1　ASP 内置组件

组　件　名	描　　述
Ad Rotator	创建一个 Ad Rotator 对象，按用户指定的播放顺序，在同一页上自动轮换显示各种广告
Browser Capabilities	创建一个 Browser Type 对象，该对象决定访问 Web 站点的每个浏览器的性能、类型及版本
Database Access	提供用 ADO 对数据库的访问
Content Linking	创建一个 Next Link 对象，该对象可动态生成 Web 页内容列表，并像书一样将各页顺序连接
File Access	提供对 Web 服务器端文件的输入/输出访问

5.2　Ad Rotator 组件

使用 Ad Rotator 组件能很容易实现在 Web 页面上插入一个循环播放广告横幅的播放器，可以通过定义一个播放顺序文件使播放器根据文件中定义的播放顺序轮流播放一系列广告图片。

Ad Rotator 组件所使用的播放顺序文件是一个文本文件，只要修改这个文件，就可以实现广告横幅播放顺序的更新。

一般来说，Ad Rotator 组件相关文件除了必要的 Adrot.dll 文件外，还要有循环定时文件，另外也可以有重定向文件。

语法如下：

```
Set AdRotator 变量 = Server.CreateObject("MSWC.AdRotator")
AdRotator 变量.属性 = 属性值 或 AdRotator 变量.方法(循环定时文件路径)
```

5.2.1　Ad Rotator 组件的属性和方法

Ad Rotator 组件的属性和方法的描述如表 5-2 所示。

表 5-2　Ad Rotator 组件属性和方法

属 性 名	语　法	描　述
Border	Border=size	允许用户指定显示广告时四周是否带边框。size 指定显示的广告四周的边框宽度，其默认值在 Rotator Schedule 文件的文件头中设置
Clickable	Clickable=value	允许用户指定是否将广告作为超链接显示。value 指定广告是否为超链接。参数值为布尔值：当为 TRUE，将广告作为超链接显示；当为 FALSE，将广告不作为超链接显示。默认值是 TRUE
TargetFrames	TargetFrames=frame	用于指定链接将被装入的目标框架。该属性完成的功能等价于 HTML 语句中的 TARGET 参数。frame 指定用来显示广告框架的名称或 HTML 框架关键字，默认值是 NO FRAME

【例 5-1】　利用 Ad Rotator 组件制作广告(如图 5-1 所示)。

【代码】

```
<HTML>
<HEAD><TITLE>Ad Rotator 组件
    </TITLE></HEAD>
<BODY bgcolor="#00ffbb">
 <CENTER><H3>Ad Rotator 组件的使用</H3></CENTER>
 <% Set ads = Server.CreateObject("MSWC.AdRotator")
        ads.border="0"
        ads.clickable=true
```



图 5-1　Ad Rotator 组件的使用

```
  %>
 <CENTER><% = ads.getadvertisement("4-1.txt")%></CENTER>
</BODY>
</HTML>
```

5.2.2 Ad Rotator 组件相关文件

1. 循环定时文件

循环定时文件用于管理和显示各种广告图像的信息。文件格式如下：

```
[REDIRECT URL]
[WIDTH numWidth]
[HEIGHT numHeight]
[BORDER numBorder]
 *
adURL
adHomePageURL
Text
Impressions
```

2. 重定向文件

重定向文件是 Ad Rotator 组件中的可选项，如果在循环定时文件中指定使用重定向文件，每当用户单击广告时，其发出的请求对象会重定向到该文件。重定向文件是使用 Request.querystring("URL")来取得与被单击图像相关的"URL"的。

该组件只有 GetAdvertisement 一个方法，用于重新取得一个将要显示的图片和超文本链接。当用户第一次启动网页或重新刷新时，系统会重新取得下一个图片。该方法的语法格式如下：

```
GetAdvertisement(rotationSchedulePath)
```

其中，rotationSchedulePath 指定 Rotator Schedule 文件相对于虚拟目录的位置，例如：

```
MyAd.GetAdvertisement("myadrot.txt")
```

5.3 Browser Capabilities 组件

此组件用来获取客户端浏览器的类型。

工作原理如下：当客户的浏览器连接到服务器时，会自动向服务器发送一个用户代理(User Agent)的 HTTP 标头(header)，该标头是一个声明浏览器及其版本的 ASCII 字符串。可以使用如下代码获得这段 header：

```
<% Request.ServerVariables("HTTP_USER_AGENT")%>
```

并将取得的用户代理 HTTP 头字符串与 Browscap.ini 文件内容进行比较，若找到匹配项，根据 Browscap.ini 文件中的定义就可以确定该浏览器的特性。

组件并不能真正检测浏览器的功能，其检测的依据是 User Agent 标题和 Browscap.ini 文件中的定义。如果 Browscap.ini 文件的内容发生变化，Browser Capabilities 组件报告的浏览器能力也会相应发生变化。

Browser Capabilities 组件除包含一系列属性、方法外，还有一个外部驱动文件 Browscap.ini。语法如下：

```
Set Browser 类型变量 = Server.CreateObject("MSWC.BrowserType")
```

Browscap.ini 文件中主要定义了各种浏览器所具有的属性，可以在该文件中声明属性定义任意数量的浏览器数目，也可以设置用户浏览器不在所定义的列表中时使用的默认值。

对每一个浏览器定义，都可以提供一个用户代理的 HTTP(HTTP User Agent)标题和属性以及希望与标题相关联的值。

Browser Capabilities 组件的操作：

首先要创建一个 Browser Capabilities 的组件实例，然后再通过该组件实例判断浏览器的类型及相关特性。

【例 5-2】 Browser Capablities 组件的使用(如图 5-2 所示)。



图 5-2　Browser Capablities 组件的使用

```
<HTML>
<HEAD><TITLE>Browser Capablities 组件</TITLE></HEAD>
<BODY bgcolor = "#00FFDD">
<% Set bc = server.createobject("MSWC.BrowserType") %>
<CENTER>Browser Capablities 组件的使用</CENTER>
<HR color = "#7A67EE">
 <TABLE align = "center" width = "180" border = "1" bordercolor = "red">
<TR><TD width = "120">浏览器名称</TD><TD><% = bc.browser %></TD></TR>
<TR><TD>浏览器版本</TD><TD><% = bc.version %></TD></TR>
<TR><TD>浏览器平台</TD><TD><% = bc.platform %></TD></TR>
<TR><TD>是否支持 Frames </TD><TD><% if (bc.frames = TRUE) then %>支持<% else %>不支持
    <% end if %></TD></TR>
<TR><TD>是否支持 Tables </TD><TD><% if (bc.tables = TRUE) then %>支持<% else %>不支持
    <% end if %></TD></TR>
<TR><TD>是否支持 VBScript </TD><TD><% if (bc.vbscript = TRUE) then %>支持<% else %>不支
    持<% end if %></TD></TR>
<TR><TD>是否支持 JScript </TD><TD><% if (bc.javascript = TRUE) then %>支持<% else %>不
    支持<% end if %></TD></TR>
<TR><TD>是否支持 Cookies </TD><TD><% if (bc.cookies = TRUE) then %>支持<% else %>不支持
    <% end if %></TD></TR>
<TR><TD>是否支持背景音乐</TD><TD><% if (bc.backgroundsounds = TRUE) then %>支持<%
    else %>不支持<% end if %></TD></TR>
</TABLE>
```

```
<% set bc = nothing %>
</BODY>
</HTML>
```

5.4 Content Linking 组件

Content Linking 组件的主要目的在于管理网页或网址间的超文本链接，可以通过一个网页或网址的线性排列顺序列表来管理多个网页或网址间的超文本链接顺序。

Content Linking 组件可创建一个管理 URL 列表的 Nextlink 对象。其语法如下：

```
Set NextLink =
Server.CreateObject("MSWC.NextLink")
```

1. 线性排列顺序文件

线性排列顺序文件包含 Web 页的相关信息列表，该文件必须在 Web 服务器的虚拟路径上可用。

语法如下：

```
Web-page-URL [ text-description [ comment]]
```

- Web-page-URL 是以 filename 或 directory\filename 表示的 Web 页的虚拟或相对 URL。
- text-description 是 Web-page-URL 的描述文字。
- comment 为组件不处理时的说明性文字。

2. Content Linking 组件方法（如表 5-3 所示）

表 5-3 Content Linking 组件方法

方法名	语法	描述
GetListCount	GetListCount(listURL)	统计内容链接列表文件中链接的项目数
GetNextURL	GetNextURL(listURL)	获取内容链接列表文件中所列的下一页的 URL
GetPreviousDescription	GetPreviousDescription(listURL)	获取内容链接列表文件中所列的上一页的说明行
GetListIndex	GetListIndex(listURL)	获取内容链接列表文件中当前页的索引
GetNthDescription	GetNthDescription(listURL, i)	获取内容链接列表文件中所列的第 i 页的 URL
GetPreviousURL	GetPreviousURL(listURL)	获取内容链接列表文件中所列的上一页的 URL
GetNextDescription	GetNextDescription(listURL)	获取内容链接列表文件中所列的下一页的 URL
GetNthURL	GetNthURL(listURL, i)	获取内容链接列表文件中所列的第 i 页的 URL

【例 5-3】 Content Linking 组件的使用(如图 5-3 所示)。

```
<HTML>
<%
   set mylink = Server. CreateObject ("MSWC.Nextlink")
   count = mylink.GetlistCount("5 - 4.txt")
   i = 1
   response.write "<CENTER>Content Linking 组件的使用</CENTER>"
 %>
<TABLE align = center>
<% Do While i <= count %>
<TD>
   <A HREF = "<% = mylink.GetNthURL("5 - 4.txt",i) %>">
   <% = MYLINK.GetNthDescription("5 - 4.txt",i) %></A>
</TD>
<%  i = i + 1 Loop %>
</TABLE></HTML>
```

Content Linking组件的使用
第一章 概述 第二章 过程 第三章 结果

图 5-3 Content Linking 组件的使用

5.5 计数器组件

在一个计数器组件中可以有多个不同名称的计数器,利用这些不同的计数器可以统计网站中多个元素的数据。

计数器组件的创建语法如下:

```
Set MyCount = Server.CreateObject("MSWC.Counters")
```

1. 计数器组件的方法(如表 5-4 所示)

表 5-4 计数器组件方法

方法名	语法	描述
Get	Get(CounterNumber)	返回指定的计数器值,如果指定的计数器不存在,则创建并将其值设为 0
Increment	Increment(CounterNumber)	对指定的计数器进行加 1 操作,如果指定的计数器不存在,则创建并将其值设为 1
Remove	Remove(CounterNumber)	清除指定的计数器
Set	Set(CounterNumber,integer)	设置指定计数器的数值

2. 计数器组件的使用

ASP 中的计数器组件通常在 Global.asa 文件中声明,如下所示:

```
<OBJECT RUNAT = Server
SCOPE = Application ID = MyCount
PROGID = "MSWC.Counters">
</OBJECT>
```

在 Global.asa 文件中利用 HTML 的<OBJECT>标记创建一个名为 MyCount 的计数器实例。

【例 5-4】 计数器组件的使用(如图 5-4 所示)。

```
Counter1.asp:
最有潜力明星投票结果统计如下:<P>
<%
xuanpiao = Request.Form("xuanpiao")
select case xuanpiao
case "chen"
'将计数器 chencounter 的值加 1,如果不存在则创建它,并将值置为 1
Counters.Increment("chencounter")
case "dong"
'将计数器 dongcounter 的值加 1,如果不存在则创建它,并将值置为 1
Counters.Increment("dongcounter")
'将计数器 lincounter 的值加 1,如果不存在则创建它,并将值置为 1
case "lin"
Counters.Increment("lincounter")
end select
%>
姚明得票数:
<% = Counters.get("chencounter") %><BR>
刘翔得票数:<% = Counters.get("dongcounter") %><BR>
田亮得票数:<% = Counters.get("lincounter") %>

Counter2.asp
<HTML>
<P>访问此页面的统计:</P>
<% num = Request("num")
if num = "" then %>
按 F5 刷新页面,计数器 pageHits 的值是:<% = counters.Increment("pageHits") %>
<% elseif num = "remove" then
counters.Remove("pageHits")
%>
计数器 pageHits 已被删除,并通过 Get 方法重新创建它,并显示它的初始值:<% = counters.get("
pageHits") %>
<% else
Call counters.set ("pageHits",Cint(num) )
%>
通过 Set 方法将页面访问计数器的值设置为固定值:<% = counters.get("pageHits") %>
<% end if %>
<FORM method = "post" action = "counter2.asp">
设置固定数值:<input type = "text" name = "num"><BR>
<INPUT type = "submit" value = "设置">
<INPUT type = "button" onclick = "javascript:window.location = 'counter2.asp?num = remove'"
```

```
value = "删除计数器">
    <INPUT type = "button" onclick = "javascript:window.location = 'co
    unter2.asp'" value = "刷新页面">
    </FORM>

    Count.htm
    <HTML>
    <HEAD>
    <TITLE>选择最有潜力明星</TITLE></HEAD>
    <body>
    从下面候选人当中选择一个最有潜力的明星:<P>
<FORM method = "post" action = "counter1.asp">
<INPUT type = radio name = "xuanpiao" value = "chen">姚明
<INPUT type = radio name = "xuanpiao" value = "dong">刘翔
<INPUT type = radio name = "xuanpiao" value = "lin">田亮
<INPUT type = "submit" value = "投上一票">
</FORM>
</BODY>
</HTML>
```

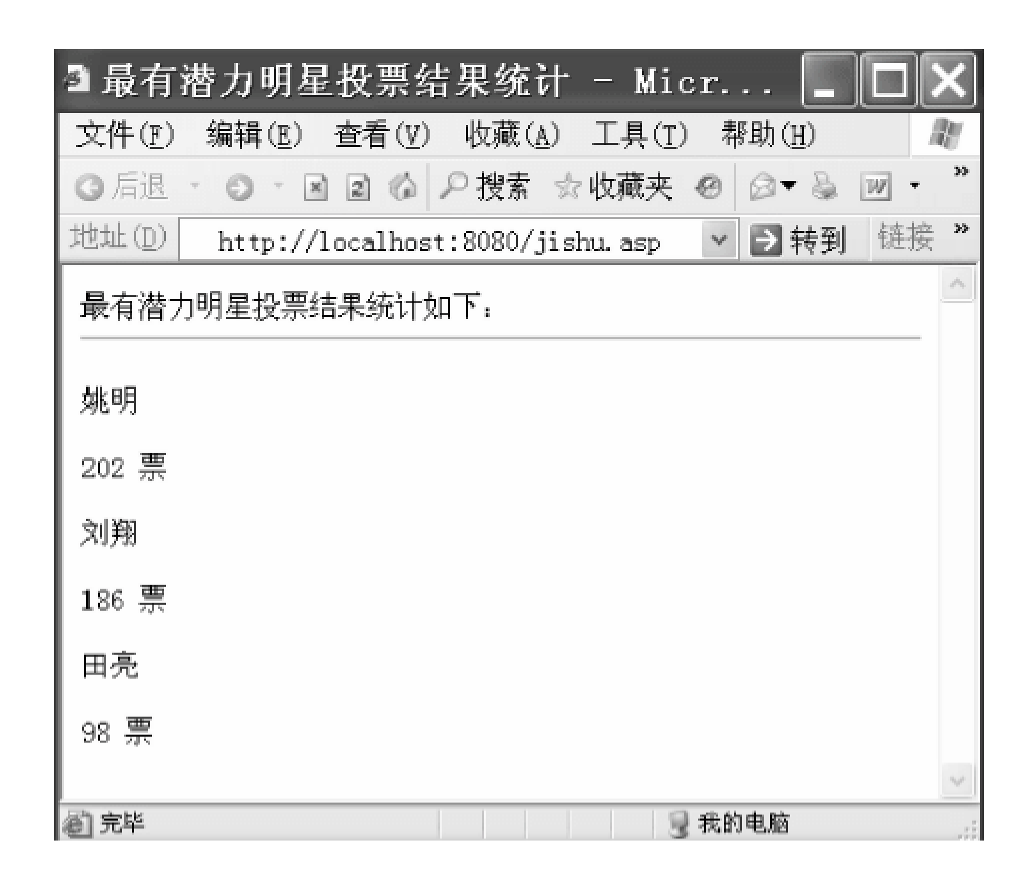


图 5-4 计数器组件的使用

5.6 CDONTS 组件

采用 CDONTS 组件在 ASP 中实现电子邮件的功能。CDONTS 组件属性如表 5-5 所示，CDONTS 组件方法如表 5-6 所示。

表 5-5 CDONTS 组件属性

属 性	说 明
To	设置收件人的邮件地址
Ac,Bcc	设置一个或多个参阅的邮件地址，是用于设置抄送人与密件抄送人的邮件地址
From	设置发件人的邮件地址
Subject	设置邮件主题
Body	设置邮件内容

表 5-6 CDONTS 组件方法

方 法	说 明	方 法	说 明
Send	发送邮件	AttachFile	设置附加文件

CDONTS 组件是 IIS 4.0 或 IIS 5.0 中提供的一种发送 E-mail 的组件，其实例的创建也要使用 Server 对象的 CreateObject 方法。其语法如下：

```
Set myMail = Server.CreateObject("CDONTS.NewMail")
```

在使用前，保证在计算机中有 CDONTS.DLL 文件（该文件一般存放在 Winnt\System32 目录下）。

首先使用 DOS 下的“regsvr32”命令注册该组件。注册该组件的语法如下：

```
regsvr32 [physical path]cdonts.dll
```

注意：应该指定 cdonts.dll 文件所在的物理路径。利用该组件提供的属性和方法进行邮件的发送。

5.7 本章小结

本章主要介绍了 ASP 常用的内置组件中的广告组件、内容组件、计数器组件等使用方法。Ad Rotator 组件可以在同一页面上自动轮换显示广告，使用 Ad Rotator 组件的步骤为：①创建 Ad Rotator 对象实例；②创建列表文件和重定向文件；③从列表文件中获取下一个广告。

Content Linking 组件是自动翻转 Web 页上的 HTML 内容字符串，使用步骤为：①创建 Content Linking 对象实例；②创建内容列表文件；③检索并显示内容字符串。

这些组件的使用为我们提供了广告推广、自动计数、自动翻页等功能，使得 ASP 的功能更加强大。

5.8 思考与练习

1. 什么是组件？
2. ASP 内置组件与 ASP 内置对象的主要区别是什么？
3. Ad Rotator 广告组件如何使用？
4. Browser Capabilities 组件如何使用？
5. 计数器组件如何使用？
6. 独立设计一个投票网站，实现投票功能。

第6章 JSP语法

JSP是一种容易学习和使用的在服务器端编译执行的Web设计语言，脚本语言采用Java，集成了Java的优点。JSP页面分为脚本元素、指令元素、动作元素。除了这3个元素外，为了增强程序的可读性，需要添加注释。

学习目标及要求

(1) 理解JSP中的注释。

(2) 熟悉JSP页面的基本结构。

(3) 掌握变量和方法的声明、Java表达式、JSP指令标签、JSP动作标签。

(4) 能够在JSP程序中使用脚本。

6.1 JSP页面的基本结构

在传统的HTML页面文件中加入Java程序片和JSP标签就构成了一个JSP页面文件。一个JSP页面可由5种元素组合而成：

(1) HTML标记符。

(2) JSP标签，如指令标签、动作标签。

(3) 变量和方法的声明。

(4) Java程序片。

(5) Java表达式。

称(3)、(4)、(5)形成的部分为JSP的脚本部分。

当服务器上的一个JSP页面被第一次请求执行时，服务器上的JSP引擎首先将JSP页面文件翻译成一个Java文件，再将这个Java文件编译生成字节码文件，然后通过执行字节码文件响应客户的请求。这个字节码文件的任务就是把JSP页面中普通的HTML标记符号交给客户的浏览器显示。

JSP标签、数据和方法声明、Java程序片由服务器负责执行，并将显示的结果发送给客户的浏览器。

Java表达式由服务器负责计算，并将结果转化为字符串，然后交给客户的浏览器负责显示。

在下面的例6-1中，客户通过表单向服务器提交三角形三边的长度，服务器将计算三角形的面积，并将计算的结果以及客户输入的三边长度返回给客户。

【例 6-1】 计算三角形面积(如图 6-1 所示)。

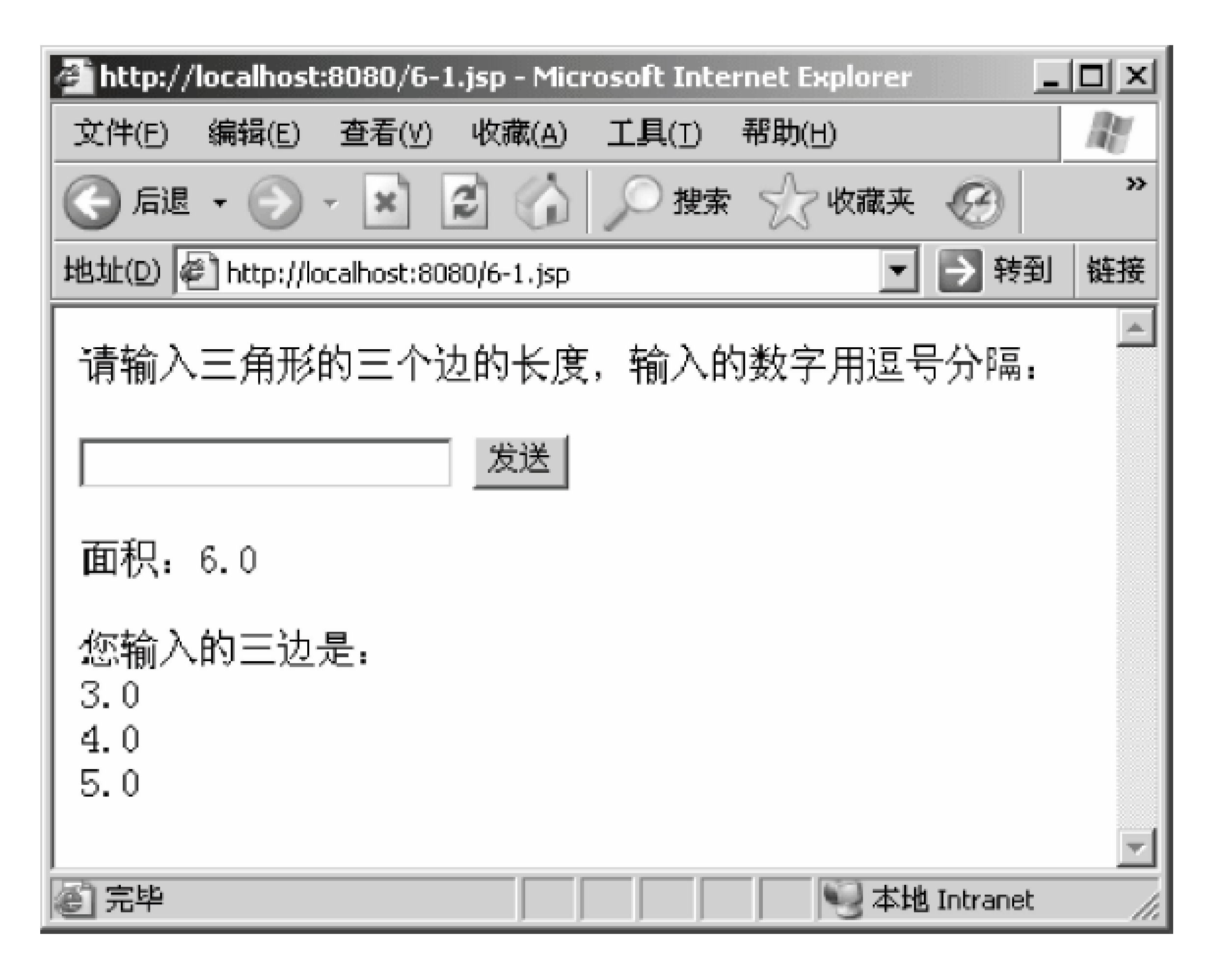


图 6-1 计算三角形的面积

```
<%@ page contentType = "text/html;charset = GB2312" %>
<%@ page import = "java.util.*" %>
 <HTML>
 <BODY BGCOLOR = cyan><FONT Size = 1>
   <P>请输入三角形的三个边的长度,输入的数字用逗号分隔:
   <BR>
   <FORM action = "Example2_1.jsp" method = post name = form>
   <INPUT type = "text" name = "boy">
   <INPUT TYPE = "submit" value = "送出" name = submit>
    </FORM>
        <%! double a[] = new double[3]
            String answer = null;
         %>
        <% int i = 0;
           boolean b = true;
           String s = null;
           double result = 0;
           double a[] = new double[3];
           String answer = null;
           s = request.getParameter("boy");
               if(s! = null)
                  { StringTokenizer  fenxi = new StringTokenizer(s,",,");
                    while(fenxi.hasMoreTokens())
                         { String temp = fenxi.nextToken();
                           try{ a[i] = Double.valueOf(temp).doubleValue();
                                     i++;
                               }
                          catch(NumberFormatException e)
                              {out.print("<BR>" + "请输入数字字符");
                              }
                         }
```

```
(32)                if(a[0] + a[1]> a[2]&&a[0] + a[2]> a[1]&&a[1] + a[2]> a[0]&&b == true)
(33)                    { double p = (a[0] + a[1] + a[2])/2;
(34)                      result = Math.sqrt(p * (p - a[0]) * (p - a[1]) * (p - a[2]));
(35)                      out.print("面积: " + result);
(36)                    }
(37)                else
(38)                    {answer = "您输入的三边不能构成一个三角形";
(39)                      out.print("<BR>" + answer);
(40)                    }
(41)            }
(42)        %>
(43)    <P> 您输入的三边是:
(44)        <BR>
(45)            <% = a[0] %>
(46)          <BR>
(47)            <% = a[1] %>
(48)          <BR>
(49)            <% = a[2] %>
(50)  </BODY>
(51)  </HTML>
```

解释：在该例中，(1)、(2)行是 JSP 指令标签。(3)～(10)行是 HTML 标记，其中(7)～(10)行是 HTML 表单，客户通过该表单向服务器提交数据。(11)～(13)行是数据声明部分，该部分声明的数据在整个 JSP 页面内有效。(14)～(42)行是 Java 程序片，该程序片负责计算面积，并将结果返回给客户。该程序片内声明的变量只在该程序片内有效。(45)、(47)、(49)行是 Java 表达式。

6.2 变量和方法的声明

6.2.1 声明变量

在“<%!”和“%>”标记符之间是声明变量，即在“<%!”和“%>”之间放置 Java 的变量声明语句，变量的类型可以是 Java 语言允许的任何数据类型，将这些变量称为 JSP 页面的成员变量。例如：

```
<%! int a, b = 10 , c;
    String tom = null, jerry = "love JSP";
    Date date;
%>
```

“<%!”和“%>”之间声明的变量在整个 JSP 页面内都有效，因为 JSP 引擎将 JSP 页面翻译成 Java 文件时，将这些变量作为类的成员变量。这些变量的内存空间直到服务器关闭才释放。当多个客户请求一个 JSP 页面时，JSP 引擎为每个客户启动一个线程，这些线程由 JSP 引擎服务器来管理，这些线程共享 JSP 页面的成员变量，因此任何一个用户对 JSP 页面成员变量操作的结果都会影响到其他用户。

下面的例 6-2 利用成员变量被所有用户共享这一性质，实现了一个简单的计数器。

【例 6-2】 简单的计数器(如图 6-2 所示)。



图 6-2 简单的计数器

```
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY BGCOLOR = cyan><FONT size = 1>
  <%!int i = 0;
   %>
  <%i++;
   %>
<P>您是第
  <% = i%>           //获取计数变量
  个访问本站的客户。 //显示内容:您是第几个访问本站的客户。
</BODY>
</HTML>
```

在处理多线程问题时,必须注意这样一个问题:当两个或多个线程同时访问同一个共享的变量,并且一个线程需要修改这个变量时,应对这样的问题作出处理,否则可能发生混乱。在例 6-2 中,可能发生两个客户同时请求 Example2_2.jsp 页面的情况。

6.2.2 声明方法

在"<%!"和"%>"之间声明方法,该方法在整个 JSP 页面有效,但是该方法内定义的变量只在该方法内有效。这些方法将在 Java 程序片中被调用,当方法被调用时,方法内定义的变量被分配内存,调用完毕即可释放所占的内存。当多个客户同时请求一个 JSP 页面时,他们可能使用方法操作成员变量,对这种情况应给予注意。在下面的例 6-3 中利用成员变量被所有客户共享这一特性实现客户帮助计算 π 的值(数学上有一个计算 π 的公式:$\pi/4=1-1/3+1/5-1/7+1/9-1/11+\cdots$),即每当客户访问 Example2_6.jsp 都参与一次 π 的计算。

【例 6-3】 利用页面访问次数计算 π 的值(如图 6-3 所示)。



图 6-3 利用页面访问次数计算 π 的值

```
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY bgcolor = cyan><FONT Size = 1>
  <%! double sum = 0,i = 1,j = 1;
      long number = 0;
       synchronized void JSPI()
        {   number ++ ;
          sum = sum + i/j;
           j = j + 2;
           i = - i;
        } %>
  <% JSPI(); %>
<P>您是第<% = number %>个访问者
<BR>
<P>目前 PI 的计算结果是:
   <% = sum * 4 %>
</BODY>
</HTML>
```

6.2.3 声明类

可以在“<%!”和“%>”之间声明一个类,该类在 JSP 页面内有效,即在 JSP 页面的 Java 程序片部分可以使用该类创建对象。在下面的例 6-4 中定义了一个 Circle 类,当客户向服务器提交圆的半径后,该类的对象负责计算面积和周长。

【例 6-4】 使用自定义的类(如图 6-4 所示)。



图 6-4 在 JSP 页面中使用自定义类

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML>
<BODY>
<P>请输入圆的半径:
<BR>
  <FORM action="Example2_7.jsp" method=get name=form>
       <INPUT type="text" name="cat" value="1">
       <INPUT type="submit" value="送出" name=submit>
  </FORM>
  <%! public class Circle
          {double r;
              Circle(double r)
               {this.r=r;
               }
            double 求面积()
               {return Math.PI*r*r;
               }
            double 求周长()
               {return Math.PI*2*r;
               }
          }
  %>
 <%  String str=request.getParameter("cat");
     double r;
     if(str!=null)
       {r=Double.valueOf(str).doubleValue();
       }
     else
       {r=1;
       }
     Circle circle=new Circle(r);         //创建对象
  %>
```

```
<P>圆的面积是:
<BR>
    <% =circle.求面积()%>
<P>圆的周长是:
<BR>
    <% =circle.求周长()%>
<HTML><BODY>
```

6.3 Java 程序片

可以在"<%"和"%>"之间插入 Java 程序片。一个 JSP 页面可以有许多程序片,这些程序片将被 JSP 引擎按顺序执行。在一个程序片中声明的变量称做 JSP 页面的局部变量,它们在 JSP 页面内的所有程序片部分以及表达式部分内都有效。这是因为 JSP 引擎将 JSP 页面翻译成 Java 文件时,将各个程序片的这些变量作为类中某个方法的变量,即局部变量。利用程序片的这个性质,有时候可以将一个程序片分割成几个更小的程序片,然后在这些小的程序片之间再插入 JSP 页面的一些其他标记元素。当程序片被调用执行时,这些变量被分配内存空间,所有的程序片调用完毕,这些变量即可释放所占的内存。当多个客户请求一个 JSP 页面时,JSP 引擎为每个客户启动一个线程,一个客户的局部变量和另一个客户的局部变量被分配不同的内存空间。因此,一个客户对 JSP 页面局部变量操作的结果,不会影响到其他客户的这个局部变量。

下面例 6-5 中的程序片负责计算 1 到 100 的连续和。

【例 6-5】 计算 1 到 100 的连续和(如图 6-5 所示)。



图 6-5 使用程序片计算连续和

```
<%@ page contentType="text/html;charset=GB2312" %>
<HTML>
<BODY bgcolor=cyan><FONT size=1>
  <%!
      long continueSum(int n)
          { int sum=0;
              for(int i=1;i<=n;i++)
                { sum=sum+i;
                }
            return sum;
          }
```

```
    %>
<P>1 到 100 的连续和:<BR>
  <% long sum;
       sum = continueSum(100);
       out.print("" + sum);
   %>
</BODY>
</HTML>
```

下面的例 6-6 将例 6-5 中的程序片分割成几部分。

【例 6-6】 程序片分割(如图 6-6 所示)。

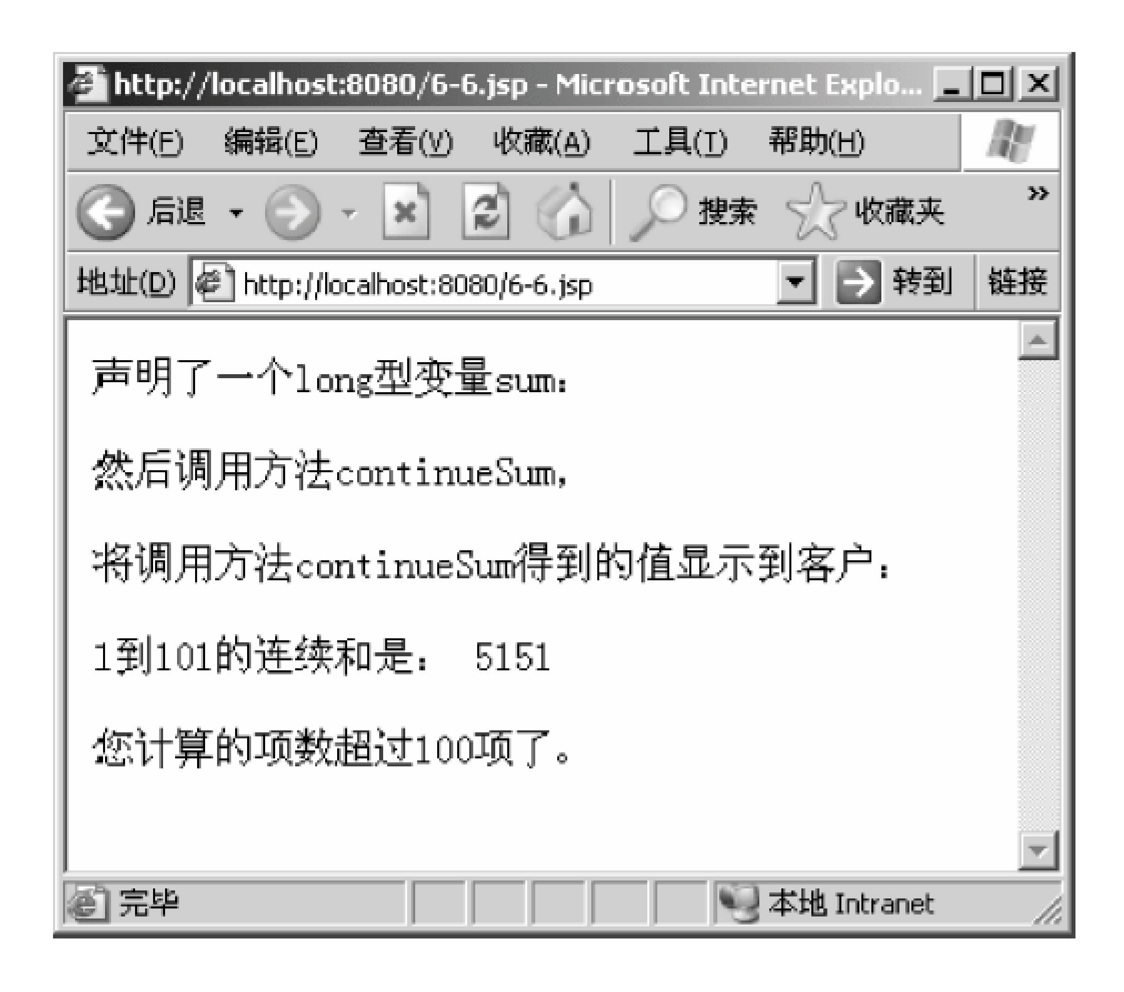


图 6-6 程序片分割

```
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY bgcolor = cyan><FONT size = 1>
   <%!
       long continueSum(int n)
            { int sum = 0;
                 for(int i = 1;i<= n;i++)//使用 for 循环,依次将变量求和,保存在 sum 中
                   { sum = sum + i; }
              return sum; }   %>
<P>声明了一个 long 型变量 sum:
   <% long sum; %>
<P>然后调用方法 continueSum,
   <% sum = continueSum(101);   %>
<P>将调用方法 continueSum 得到的值显示到客户:
<BR>
<P>1 到 101 的连续和是:
   <% = sum   %>
 <% if(sum >= 5050)
   { %>   <P>您计算的项数超过 100 项了。<% }
   else
  { %>  <P>您计算的项数没有超过 100 项。<%
   }   %>
</FONT></BODY></HTML>
```

下面例 6-7 的程序片负责读取服务器上的一个文件，并将文件的内容显示给客户。

【例 6-7】 使用程序片读取文件(如图 6-7 所示)。

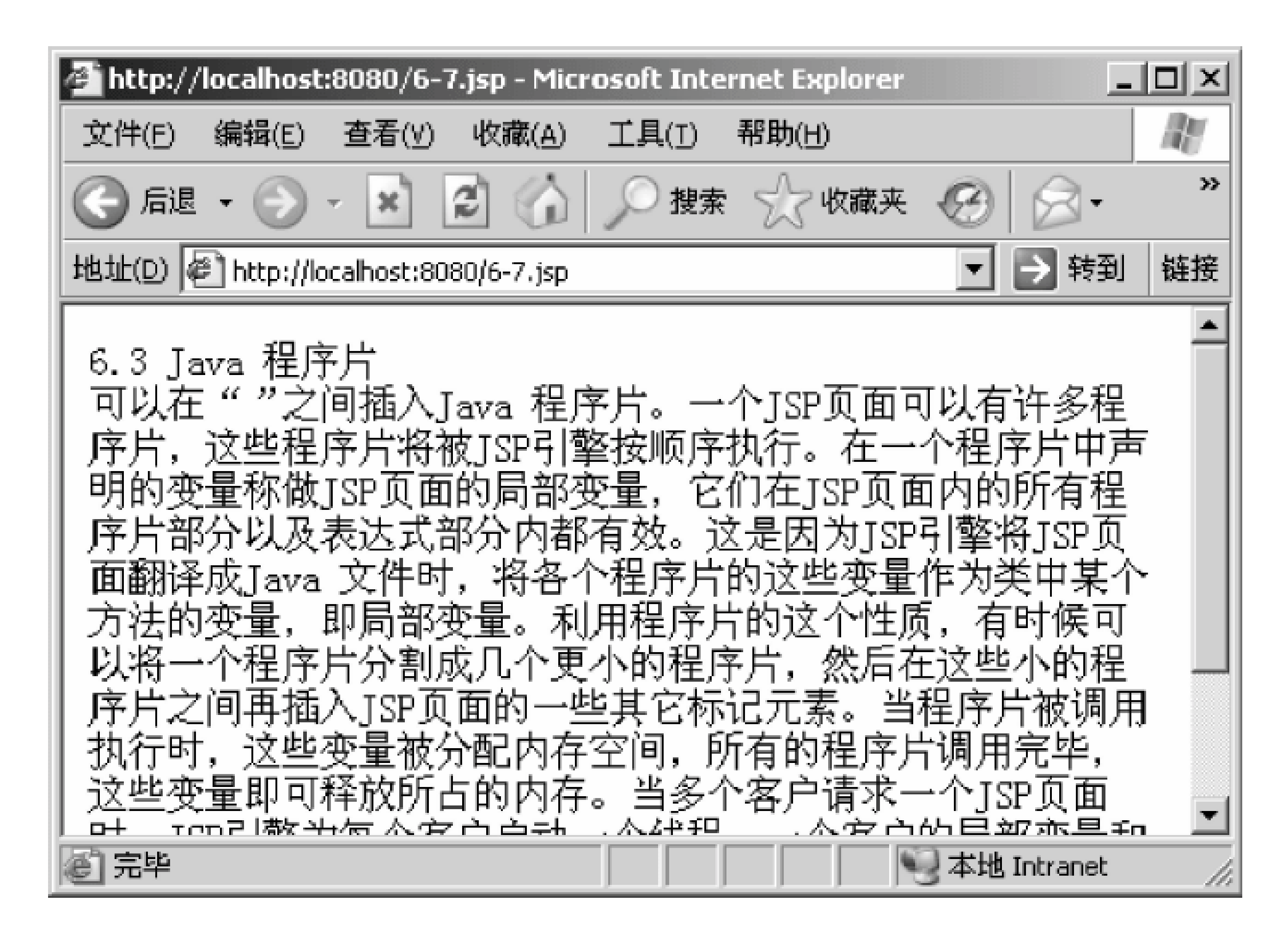


图 6-7　使用程序片读取文件

```
<%@ page contentType="text/html;charset= GB2312" %>
<%@ page import="java.io.*" %>
<%@ page contentType="text/html;charset=GB2312" %>
<%@ page import="java.io.*" %>
<HTML>
<BODY bgcolor=cyan><FONT size=1>
 <% try{ File f=new File("D:/test","A.txt");
         FileReader in=new FileReader(f);
         BufferedReader buffer=new BufferedReader(in);
         String s=null;
         while((s=buffer.readLine())!=null)
             { out.print(s+"<BR>"); }
         in.close();
         buffer.close(); }
    catch(IOException ee)
        {out.print("文件不存在");} %>
</BODY>
</HTML>
```

6.4 Java 表达式

可以在“<%=”和“%>”之间插入一个表达式(注意：不可插入语句，“<%=”是一个完整的符号，“<%”和“=”之间不要有空格)，这个表达式必须能求值。表达式的值由服务器负责计算，并将计算结果用字符串形式发送到客户端显示。

下面的例 6-8 计算表达式的值。

【例 6-8】 计算表达式的值(如图 6-8 所示)。

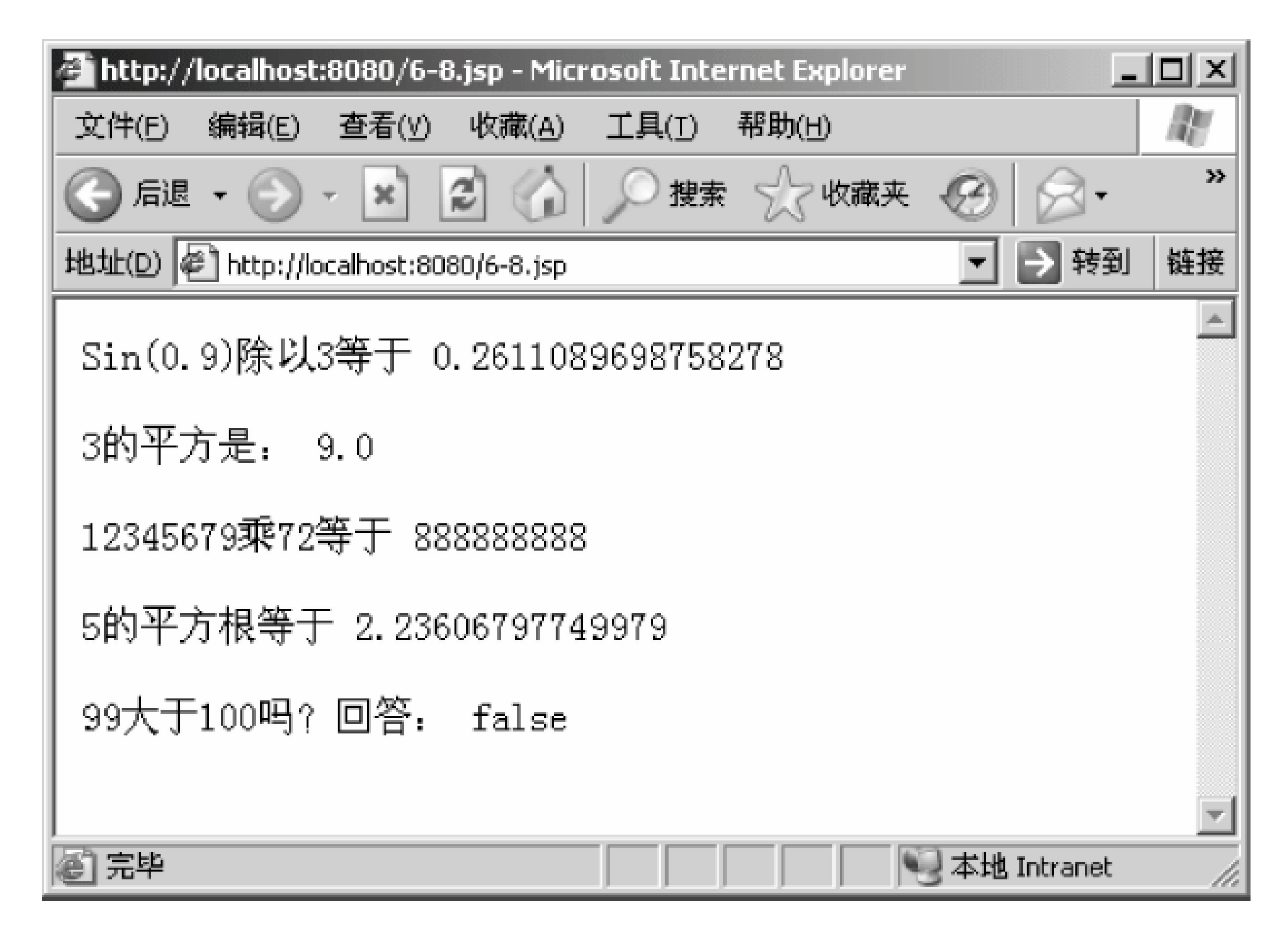


图 6-8 计算表达式的值

```
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY  bgcolor = cyan><FONT size = 1>
<P>Sin(0.9)除以 3 等于
  <% = Math.sin(0.90)/3 %>
<p>3 的平方是:
  <% = Math.pow(3,2) %>
<P>12345679 乘 72 等于
  <% = 12345679 * 72 %>
<P>5 的平方根等于
  <% = Math.sqrt(5) %>
<P>99 大于 100 吗?回答:
  <% = 99 > 100 %>
</BODY>
</HTML>
```

6.5 JSP 中的注释

注释可以增强 JSP 文件的可读性,并易于 JSP 文件的维护。JSP 中的注释可分为以下两种。

(1) HTML 注释:在标记符号“<! ——”和“——>”之间加入注释内容。格式如下:

```
<! --   注释内容   -->
```

JSP 引擎把 HTML 注释交给客户,因此客户通过浏览器查看 JSP 的源文件时,能够看到 HTML 注释。

(2) JSP 注释:在标记符号“<%——”和“——%>”之间加入注释内容。格式如下:

```
<% --   注释内容   -- %>
```

JSP 引擎忽略 JSP 注释,即在编译 JSP 页面时忽略 JSP 注释。

6.6 JSP 指令标签

6.6.1 page 指令标签

page 指令标签用来定义整个 JSP 页面的一些属性和这些属性的值。例如，可以用 page 指令标签定义 JSP 页面的 contentType 属性的值是“text/html;charset=GB2312”，这样，页面就可以显示标准汉语。如：

```
<%@ page contentType = "text/html;charset = GB2312" %>
```

page 指令标签的格式如下：

```
<%@ page  属性 1 = "属性 1 的值"  属性 2 = "属性 2 的值"  … %>
```

属性值总是用单引号或双引号括起来，例如：

```
<%@ page contentType = "text/html;charset = GB2312" import = "java.util. * " %>
```

如果为一个属性指定几个值的话，这些值用逗号分隔。page 指令标签只能给 import 属性指定多个值；其他属性只能指定一个值。

例如：

```
<%@ page  import = "java.util. * " ,"java.io. * " , "java.awt. * " %>
```

当为 import 指定多个属性值时，JSP 引擎把 JSP 页面翻译成的 Java 文件中会有如下的 import 语句：

```
import java.util. * ;
import java.io. * ;
import java.awt. * ;
```

在一个 JSP 页面中，也可以使用多个 page 指令标签来指定属性及其值。需要注意的是：可以使用多个 page 指令标签给属性 import 几个值，但其他属性只能使用一次 page 指令标签指定该属性的一个值。如：

```
<%@ page contentType = "text/html;charset = GB2312" %>
<%@ page import = "java.util. * " %>
<%@ page import = "java.util. * ", "java.awt. * " %>
```

注：下列用法是错误的：

```
<%@ page contentType = "text/html;charset = GB2312" %>
<%@ page contentType = "text/html;charset = GB2312" %>
```

尽管指定的属性值相同，也不允许两次使用 page 给 contentType 属性指定属性值。

注：page 指令标签的作用对整个页面有效，与其书写的位置无关，但习惯把 page 指令标签写在 JSP 页面的最前面。

下面给出 page 的几个常用属性。

1）language 属性

定义 JSP 页面使用的脚本语言，该属性的值目前只能取“java”。

为 language 属性指定值的格式如下：

```
<%@ page  language="java" %>
```

language 属性的默认值是“java”，即如果你在 JSP 页面中没有使用 page 指令标签指定该属性的值，那么 JSP 页面默认有如下 page 指令：

```
<%@ page language="java" %>
```

2）import 属性

该属性的作用是为 JSP 页面引入 Java 核心包中的类，这样就可以在 JSP 页面的程序片部分、变量及函数声明部分、表达式部分使用包中的类。可以为该属性指定多个值，该属性的值可以是 Java 某包中的所有类或一个具体的类，例如：

```
<%@ page import="java.io.*", "java.util.Date" %>
```

JSP 页面默认 import 属性已经有如下的值：“java.lang.*”、“javax.servlet.*”、“javax.servlet.jsp.*”、“javax.servlet.http.*”。

3）contentType 属性

该属性定义 JSP 页面响应的 MIME(multipurpose Internet mail extention)类型和 JSP 页面字符的编码。属性值的一般形式是：“MIME 类型”或“MIME 类型；charset=编码”，如：

```
<%@ page contentType="text/html;charset=GB2312" %>
```

contentType 属性的默认值是“text/html ；charset=ISO-8859-1”。

4）session 属性

该属性用于设置是否需要使用内置的 session 对象。

session 的属性值可以是 true 或 false，session 默认的属性值是 true。

5）buffer 属性

内置输出流对象 out 负责将服务器的某些信息或运行结果发送到客户端显示，buffer 属性用来指定 out 设置的缓冲区的大小或不使用缓冲区。

buffer 属性可以取值 none，设置 out 不使用缓冲区。buffer 属性的默认值是 8Kb。例如<%@ page buffer= “24Kb” %>。

6）autoFlush 属性

指定 out 的缓冲区被填满时，缓冲区是否自动刷新。

autoFlush 可以取值 true 或 false。autoFlush 属性的默认值是 true。当 autoFlush 属性取值 false 时，如果 out 的缓冲区填满时，就会出现缓存溢出异常。当 buffer 的值是 none 时，autoFlush 的值就不能设置成 false。

7）isThreadSafe 属性

该属性用来设置 JSP 页面是否可多线程访问。

isThreadSafe 的属性值可以取值 true 或 false。当 isThreadSafe 属性值设置为 true 时，JSP 页面能同时响应多个客户的请求；当 isThreadSafe 属性值设置成 false 时，JSP 页面同一时刻只能响应一个客户的请求，其他客户须排队等待。isThreadSafe 属性的默认值是 true。

8）info 属性

该属性为 JSP 页面准备一个字符串，属性值是某个字符串。例如：

```
<%@ page info= "we are students" %>
```

可以在 JSP 页面中使用以下方法获取 info 属性的属性值：

```
getServletInfo();
```

注：当 JSP 页面被翻译成 Java 文件时，翻译成的类是 Servlet 的一个子类，所以在 JSP 页面中可以使用 Servlet 类的方法 getServletInfo()。

6.6.2　include 指令标签

如果需要在 JSP 页面内某处整体嵌入一个文件，就可以考虑使用 include 指令标签。该指令标签语法如下：

```
<%@ include file= "文件的名字" %>
```

该指令标签的作用是在 JSP 页面出现该指令的位置处静态插入一个文件。被插入的文件必须是可访问和可使用的，即该文件必须和当前 JSP 页面在同一 Web 服务目录中。所谓静态插入，就是当前 JSP 页面和插入的部分合并成一个新的 JSP 页面，然后 JSP 引擎再将这个新的 JSP 页面翻译成 Java 类文件。因此插入文件后，必须保证新合并成的 JSP 页面符合 JSP 语法规则，即能够成为一个 JSP 页面文件。比如，如果一个 JSP 页面使用 include 指令插入另一个 JSP 文件，被插入的这个 JSP 页面中有一个设置页面 contentType 属性的 page 指令<%@ page contentType="text/html;charset=GB2312" %>，而当前 JSP 页面已经使用 page 指令设置了 contentType 的属性值，那么新合并的 JSP 页面就出现了语法错误，当翻译合并的 JSP 页面到 Java 文件时就会失败。

下面的例 6-9 在 JSP 页面静态插入一个文本文件 Hello.txt，该文本文件的内容是：“你们好，很高兴认识你们呀！”该文本文件必须和当前 JSP 页面在同一 Web 服务目录中。

【例 6-9】　使用 include 指令标签静态嵌入文本文件（如图 6-9 所示）。



图 6-9　使用 include 指令标签静态嵌入文本文件

```
<%@ page contentType = "text/html;charset = GB2312" %>
<html>
<BODY bgcolor = cyan>
<H3>
  <%@ include file = "Hello.txt" %>
```

```
</H3>
</BODY>
</HTML>
```

注：上述 Example2_14.jsp 等价于下面的 JSP 文件 Example2_14_1.jsp。

```
Example2_14_1.jsp:
<%@ page contentType="text/html;charset=GB2312" %>
<html>
<BODY>
<H3>
你们好,很高兴认识你们呀!
</H3>
</BODY>
</HTML>
```

注：在 Example2_14.jsp 被翻译成 Java 文件后，如果对插入的文件 Hello.txt 进行了修改，那么必须重新将 Example2_14.jsp 翻译成 Java 文件（重新保存页面，然后再访问该页面即可），否则只能看到修改前的 Hello.txt 的内容。

【例 6-10】 在 JSP 页面中静态插入一个 JSP 文件 Computer.jsp。

```
Computer.jsp:
<FORM action="" method=post name=form>
<INPUT type="text" name="ok">
<BR><INPUT TYPE="submit" value="送出" name=submit>
</FORM>
  <% String a=request.getParameter("ok");
  if(a==null)
    {   a="1";
    }
  try{
      double number=Integer.parseInt(a);
  out.print("<BR>"+Math.sqrt(number));
    }
     catch(NumberFormatException e)
    {
    out.print("<BR>"+"请输入数字字符"); }   %>
```

【例 6-11】 使用 include 指令标签静态嵌入 JSP 文件（如图 6-10 所示）。



图 6-10 使用 include 指令标签静态嵌入 JSP 文件

```
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY Bgcolor = cyan><FONT size = 1>
<P>请输入一个正数,点击按钮求这个数的平方根。
<CENTER>
<%@ include file = "Computer.jsp" %>
</CENTER>
</BODY>
</HTML>
```

注：上述 JSP 文件等同于下述 JSP 文件：

```
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY Bgcolor = cyan><FONT size = 1>
<P>请输入一个正数,点击按钮求这个数的平方根。
<CENTER>
<FORM action = "" method = post name = form>
<INPUT type = "text" name = "ok">
<BR><INPUT type = "submit" value = "送出" name = submit>
</FORM>
  <%  String a = request.getParameter("ok");
     if(a == null)
       {  a = "1";  }
    try{ double number = Integer.parseInt(a);
         out.print("<BR>" + Math.sqrt(number)); }
    catch(NumberFormatException e)
         { out.print("<BR>" + "请输入数字字符"); } %></CENTER></BODY></HTML>
```

6.7 JSP 动作标签

动作标签是一种特殊的标签,它影响 JSP 运行时的功能。

6.7.1 include 动作标签

include 动作标签格式如下：

```
<jsp:include page = "文件的名字"/>
```

或

```
<jsp:include page = "文件的名字"></jsp:include>
```

该动作标签告诉 JSP 页面动态包含一个文件,即 JSP 页面运行时才将文件加入。与静态插入文件的 include 指令标签不同,当 JSP 引擎把 JSP 页面翻译成 Java 文件时,不把 JSP 页面中动作指令 include 所包含的文件与原 JSP 页面合并成一个新的 JSP 页面,而是告诉 Java 解释器这个文件在 JSP 运行时(Java 文件的字节码文件被加载执行)才包含进来。如

果包含的文件是普通的文本文件，就将文件的内容发送到客户端，由客户端负责显示；如果包含的文件是 JSP 文件，JSP 引擎就执行这个文件，然后将执行的结果发送到客户端，并由客户端负责显示这些结果。

注：

(1) include 动作标签与静态插入文件的 include 指令标签有很大的不同。动作标签是在执行时才对包含的文件进行处理，因此 JSP 页面和它所包含的文件在逻辑和语法上是独立的，如果对包含的文件进行了修改，那么运行时看到的是所包含文件修改后的结果；而静态 include 指令包含的文件如果发生了变化，必须重新将 JSP 页面翻译成 Java 文件(可将该 JSP 页面重新保存，然后再访问，就可产生新的翻译 Java 文件)，否则只能看到所包含的修改前的文件内容。

(2) 书写 include 动作标签<jsp:include page…/>时要注意："jsp"、"："、"include"三者之间不要有空格。

下面的例 6-12 动态包含两个文件：image. html 和 Hello. txt。把 Example6_16. jsp 页面保存到 Tomcat\Jakarta-tomcat-4. 0\webapps\root 下，在 root 下新建一个文件夹 Myfile，将 Hello. txt 存放在 MyFile 文件夹中，将 image. html 存放在 root 下。

```
Hello.txt:
<H4>你好,祝学习进步!
<BR>学习 JSP 要有 Java 语言的基础。
<BR>要认真学习 JSP 的基本语法。
</H4>
image.html:
<image src="tom1.jpg">
```

【例 6-12】 使用 include 动作标签加载图片文件(如图 6-11 所示)。



图 6-11 用 include 动作标签加载图片文件

```
Example6_16.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY BGCOLOR = Cyan ><FONT Size = 1 >
<P>加载的文件:
   <jsp:include page = "Myfile/Hello.txt">
   </jsp:include>
<P>加载的图像:
<BR>
   <jsp:include page = "image.html">
   </jsp:include>
</BODY>
</HTML>
```

由于动作指令 include 是动态地包含一个文件,因此客户可以通过 HTML 表单提交需要包含的文件的名字。在下面的例 6-13 中,用 Java 语言中的 File 类列出了 Tomcat\Jakarta-tomcat-4.0\webapps\root\Myfile 下所有的 JSP 文件,客户通过表单,提交要加载的文件的全名。

【例 6-13】 使用 include 动作标签加载 JSP 文件(如图 6-12 所示)。

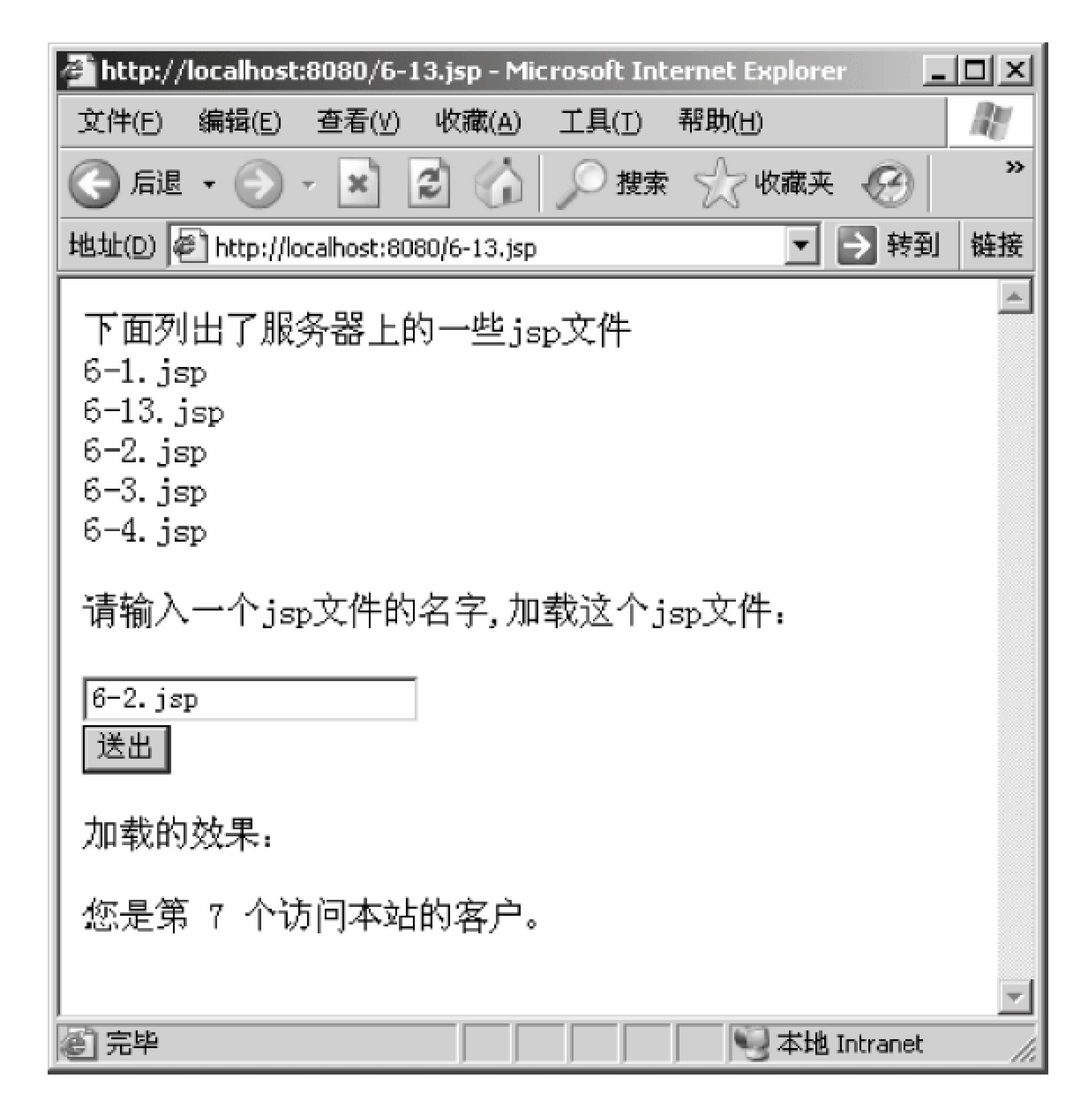


图 6-12 用 include 动作标签加载 JSP 文件

```
Example6_17.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<%@ page import  = "java.io.*" %>
<%! class FileJSP implements FilenameFilter
 { String str = null;
      FileJSP(String s)
      {str = "." + s;
```

```
        }
        public boolean accept(File dir,String
         name)
            { return name.endsWith(str);
            } } %>
<BODY bgcolor = cyan><FONT Size = 1>
<P>下面列出了服务器上的一些 JSP 文件
<% File dir = new
     File("d:/Tomcat/Jakarta - tomcat - 4.0/webapps/root/Myfile");
     FileJSP file_jsp = new FileJSP("jsp");
     String file_name[] = dir.list(file_jsp);
      for(int i = 0;i < file_name.length;i ++ )
                {out.print("<BR>" + file_name[i]); } %>
   <P>请输入一个 JSP 文件的名字,加载这个 JSP 文件:
   <FORM action = "Example2_17.jsp" method = post name = form>
   <INPUT type = "text" name = "ok">
         <BR>
        <INPUT TYPE = "submit" value = "送出" name = submit>
</FORM>
      <% String fileName = "/Myfile/";
          fileName = fileName + request.getParameter("ok"); %>
<P>加载的效果:
      <jsp:include page = "<% = fileName %>" >
      </jsp:include></BODY>
```

6.7.2 param 动作标签

param 标签以“名字-值”对的形式为其他标签提供附加信息,这个标签与 jsp:include、jsp:forward、jsp:plugin 标签一起使用。

param 动作标签格式如下:

```
<jsp:param name = "名字" value = "指定给 param 的值">
```

当该标签与 jsp:include 标签一起使用时,可以将 param 标签中的值传递到 include 指令要加载的文件中去,因此 include 动作标签如果结合 param 标签,可以在加载文件的过程中向该文件提供信息。下面的例 6-14 动态包含文件 tom.jsp,当该文件被加载时获取 param 标签中 computer 的值(获取 computer 的值由 JSP 的内置对象 request 调用 getParameter 方法完成)。

```
tom.jsp:
<% @ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY>
      <% String str = request.getParameter("computer");          //获取值
          int n = Integer.parseInt(str);
          int sum = 0;
```

```
            for(int i = 1;i<= n;i ++ )
              { sum = sum + i;
              } %><P>
        从 1 到<% = n %>的连续和是:
<BR>
     <% = sum %>
</BODY>
</HTML>
```

【例 6-14】 使用 param 标签传递参数(如图 6-13 所示)。



图 6-13 用 param 标签传递参数

```
Example6_18.jsp:
<% @ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY>
<P>加载文件效果:
      <jsp:include page = "tom.jsp">
        <jsp:param name = "computer" value = "300" /></jsp:include></BODY></HTML>
```

6.7.3 forward 动作标签

forwar 动作标签格式如下:

```
<jsp:forward page = "要转向的页面" >
   </jsp:forward>
```

或

```
<jsp:forward page = "要转向的页面" />
```

该动作标签的作用是:从该指令处停止当前页面的继续执行,而转向其他的 JSP 页面。假设 Example6_11.jsp 存放在 D:/Tomcat/Jakarta- tomcat-4.0 / webapps/root/Myfile 下;Example2_2 存放在 D:/Tomcat/Jakarta-tomcat-4.0/ webapps /root/下。下面的 Example6_19.jsp 存放在 D:/Tomcat/ Jakarta- tomcat-4.0/webapps/root/下。

在下面的 JSP 页面中,首先随机获取一个数,如果该数大于 0.5 就转向页面 Example6_

11.jsp,否则转向页面 Example6_2.jsp。图 6-14 是 Example6_19.jsp 运行几次后随机出现的两种运行效果。

【例 6-15】 使用 forward 指令进行页面转向(如图 6-14 所示)。



图 6-14 用 forward 指令进行页面转向

```
Example6_19.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY>
<% double i = Math.random();
     if(i>0.5)
     { %>
    <jsp:forward
page = "/Myfile/Example2_11.jsp" >
     </jsp:forward><% }
     else
     { %>
     <jsp:forward page = "Example2_2.jsp" >
     </jsp:forward><% } %>
<P>这句话和下面的表达式的值能输出吗?
 <% = i %></BODY>
 </HTML>
```

该指令也可以结合 param 指令,向要转到的页面传送信息。下面的 Example6_20.jsp 页面转到 come.jsp,并向转到的 come.jsp 页面传递一个数值,如图 6-15 所示。



图 6-15　用 param 标签传值到转向页面

```
come.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY bgcolor = cyan><FONT Size = 1>
<% String str = request.getParameter("number");
       double n = Double.parseDouble(str); %>
    <P>您传过来的数值是:<BR>
    <% = n %></BODY></HTML>
```

【例 6-16】 使用 param 标签传值到转向页面(如图 6-15 所示)。

```
Example6_20.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY>
 <% double i = Math.random(); %>
    <jsp:forward page = "come.jsp" >
        <jsp:param name = "number" value = "<% = i %>" />
    </jsp:forward> </BODY></HTML>
```

6.7.4　plugin 动作标签

<jsp:plugin>用于在浏览器中播放或显示一个对象(通常为 Applet 或 Bean),而显示需要在浏览器中安装一个 java 插件。当 JSP 网页被编译后送往浏览器执行时,<jsp:plugin>将会根据浏览器的版本替换成<object>标签或者<embed>标签。

一般来说,<jsp:plugin>会指定对象是 Applet 或 Bean,同样也会指定类的名字和位置,另外还会指定将从哪里下载这个 java 插件。

Plugin 动作标签格式如下:

```
<jsp:plugin type = "applet" code = "小应用程序的字节码文件"
    jreversion = "Java 虚拟机版本号" width = "小程序宽度值" height = "小程序高度值" >
    <jsp:fallback>
      提示信息: 用来提示用户的浏览器是否支持插件下载
  </jsp:fallback>
</jsp:plugin>
```

6.8 本章小结

本章介绍了 JSP 的语法规则。JSP 的语法被分为 JSP 指令、JSP 脚本元素和 JSP 的标准动作几个部分。在 JSP 页面中包含两种不同类型的注释：一种是 JSP 本身，用于描述 JSP 程序代码；另一种是 JSP 生成的页面，即 HTML 和 XML 注释。通过学习标签的内容，结合网页制作工具，编写 JSP 应用程序。

6.9 思考与练习

1. JSP 的脚本元素分为哪 3 种格式？
2. Page 指令包括哪些内容？
3. 什么是 include 指令标签？
4. 如何在 JSP 中加注释？
5. 比较 include 指令标签和 include 动作标签。
6. 利用 include 动作标签动态加载页面。
7. 利用 param 和 forward 制作转向页面。
8. 利用 JSP 技术制作网站计数器。

第7章 Servlet和JavaBean技术

随着Internet的发展，C/S模式方面的许多新技术应运而生，其中最瞩目的就是Java。Java不但是一种计算机高级语言，同时也提供了一整套客户机/服务器解决方案，在这种方案中，程序可以自动下载到客户端并执行。Servlet就是解决这种方案的最好办法。它可以被Web服务器加载和执行。JavaBean是一个可以重复使用的软件部分，该软件可以用来生成进行可视处理的组件。

学习目标及要求

(1) 理解Servlet的使用方法。

(2) 能够编写简单的Servlet。

(3) 理解JavaBean技术。

(4) 能够使用JavaBean标签。

7.1 Servlet简介

从功能上来说，Servlet与JSP完全一样。但是从编写的角度来说是不一样的：JSP是在HTML或者XML文档中嵌入Java脚本或者JSP标签形成的，是文本文件；而Servlet是纯Java文件，是一个类。所有使用Servlet完成的功能，包括与用户的交互，都是通过Java代码完成的。

Servlet是用Java语言编写的运行在服务器端的小应用程序，能够接收Web客户端的请求，并能对Web客户端进行响应，通常是通过HTTP(hypertext transfer protocol)协议进行工作的。

可以认为Servlet是服务器端的Applet(实际上，Applet运行在客户端，而Servlet运行在服务器端)。

7.1.1 Servlet的主要方法

Servlet是一个Java类，主要有3类方法，分别是：

(1) init方法，用于初始化。

(2) destroy方法，用于释放资源。

(3) service方法(服务类方法)，对用户的请求进行处理，并对用户进行响应。

此外，还有doGet和doPost等方法。doGet方法可以响应get方式的请求，doPost方

法可以响应 post 方式的请求。通常，表单提交都使用 post 方式，超链接使用 get 方式。

7.1.2 编写简单的 Servlet

1. 编写 Servlet

首先确定要实现的功能，功能确定之后，根据 Servlet 的框架来编写 Servlet。这里主要关注编写 Servlet 的一般过程。

因为 Servlet 是 Java 类，所以它与一般的 Java 类的编写基本相同。具体步骤如下：

(1) 创建一个包。假设包名是 servlets，创建包的代码如下：

```
package servlets;
```

(2) 导入用到的包。需要导入的包有 javax.servlet.* 和 javax.servlet.http.*，通常还需要导入 java.io.* 包。代码如下：

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
```

(3) 定义类的基本框架并继承 HttpServlet。每个 Servlet 都必须实现接口 Servlet，而这里要编写的 Servlet 主要用于 HTTP 协议，可以继承实现了 Servlet 接口的 HttpServlet。代码如下：

```
public class HelloServlet extends HttpServlet
{
}
```

(4) 编写成员方法。Servlet 通常不需要成员变量，成员方法包括 init、service、destroy 等。如果要对 Servlet 进行初始化，则要实现 init 方法。如果要在卸载 Servlet 时执行一些功能，则要实现 destory 方法。一般情况下，根据要完成的功能只须实现服务类方法即可。这里只实现 doGet 方法。服务类方法有两个参数：第一个参数是 HttpServletRequest 的对象，该参数包含了用户的所有请求信息，要想获取用户的请求信息，必须使用该对象提供的方法；第二个参数是 HttpServletResponse 的对象，可以通过这个对象对用户进行响应，如果希望对用户进行响应或者设置响应相关的信息，要使用该对象的方法。

【例 7-1】 本实例完成的功能是向用户输出一行文字，所以无须获取用户的请求信息，只须对用户进行响应即可。在对用户进行响应的时候，要设定文档类型以及编码方式等，就像在 JSP 中每个文件都要设置 page 标签的 contentType 属性一样。之后要向用户输出一些内容，要先获取输出流。响应页面的属性设置和输出流的获取都是与响应相关的，要使用第二个参数来完成。

【代码】

```
public void doGet(HttpServletRequest request,HttpServletResponse)
   throws IOException,ServletException
{
```

```
        response.setContentType("text/html;charset = gb2312");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>Hello World!</title></head>");
        out.println("<body>");
        out.println("<p>Hello World!</p>");
        out.println("</body></html>");
    }
```

从代码可以看出，如果使用 Servlet 完成对用户的响应，所有的响应内容，包括格式信息等，都要通过输出语句输出。如果使用 JSP 完成这个功能就非常简单了。正因为这个原因，才推出了 JSP 技术。JSP 技术的主要优势就在于与用户进行交互比较方便。所以如果要完成的功能是与用户进行交互，则应该使用 JSP 完成。

为了使这个 Servlet 能够响应 post 请求，需要实现 doPost 方法，这时只须调用 doGet 方法即可。代码如下：

```
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
   doGet(request, response);
}
```

要把这两个方法添加到前面定义的类 HelloServlet 中。

2. 编译 Servlet

Servlet 开发所需要的包有 javax.servlet.* 和 javax.servlet.http.*。这些包不是 Java 标准版的类库，如果没有安装 Java EE 开发环境，则需要配置环境变量。这些包位于 Tomcat 安装目录下的 lib 下的 servlet-api.jar 压缩包中，需要把这个压缩包添加到环境变量中。

假设 servlet-api.jar 所在的位置为 C:\Program Files\Apache Software Foundation\Tomcat 6.0\lib，需要在环境变量 classpath 中添加 C:\Program Files\Apache Software Foundation\Tomcat 6.0\lib\servlet-api.jar。

3. 配置 Servlet

Servlet 编写完之后，作为 Web 应用的组成部分，要在配置文件 web.xml 中进行配置。Servlet 的配置包括两部分：第一部分是 Servlet 的声明；第二部分是 Servlet 访问方式的设置。

首先是 Servlet 的声明，使用<servlet>元素：

```
<servlet>
  <servlet-name>hello</servlet-name>
  <servlet-class>servlets.HelloServlet</servlet-class>
</servlet>
```

<servlet-name>用于声明 Servlet 的名字，这里，hello 是 Servlet 的名字；<servlet-class>用于声明 Servlet 所对应的类，这里，servlets.HelloServlet 是这个 Servlet 所对应的类名。

然后是 Servlet 映射，主要配置访问该 Servlet 的方式，使用<servlet-mapping>元素：

```
<servlet-mapping>
  <servlet-name>hello</servlet-name>
  <url-pattern>/hello</url-pattern>
</servlet-mapping>
```

<servlet-name>表示 Servlet 的名字，要和上面的 Servlet 声明保持一致；<url-pattern>表示访问时的格式，例如要访问上面这个 Servlet 要使用 hello 作为名字访问。servlet-name 和 url-pattern 可以不相同。服务器接收到请求之后会根据这个 url-pattern 查找相应的 Servlet。

4. 部署 Servlet

Servlet 编写完成之后，需要部署到服务器才能访问，编译好的文件要放在 Web 应用的 WEB-INF\classes 下面。

5. 访问 Servlet

对 Servlet 的访问与对 JSP 的访问基本相同，可以通过以下 3 种方式进行：

(1) 通过在浏览器中直接输入地址访问：

```
http://127.0.0.1:8080/test/hello
```

(2) 通过超链接的形式访问：

```
<a href = "http://127.0.0.1:8080/test/hello">第一个 Servlet</a>
```

(3) 通过表单提交的方式访问：

```
<form method = get action = "http://127.0.0.1:8080/test/hello">
```

7.1.3 Servlet 运行原理

Servlet 的运行原理如图 7-1 所示，具体介绍如下。

(1) 服务器接收到客户端的请求时，先判断用户所请求的 Servlet 对象是否存在：

- 如果存在，则直接调用 Servlet 对象的 Service 方法(间接调用 doGet 方法或者 doPost 方法)。
- 如果不存在，则加载 Servlet 类，创建 Servlet 对象并实例化，然后调用 init 方法进行初始化，之后调用 Servlet 对象的 Service 方法。

(2) 使用 Service 方法执行的结果对用户进行响应。

在 Web 应用被卸载或者服务器被关闭时，系统卸载 Servlet，调用 destroy 方法释放资源。

Servlet 的加载有两种方式：

- 第一次请求时加载。
- 服务器启动时加载。

这两种方式各有利弊。如果第一次请求时加载，加载的速度比较慢，但是不浪费空间。

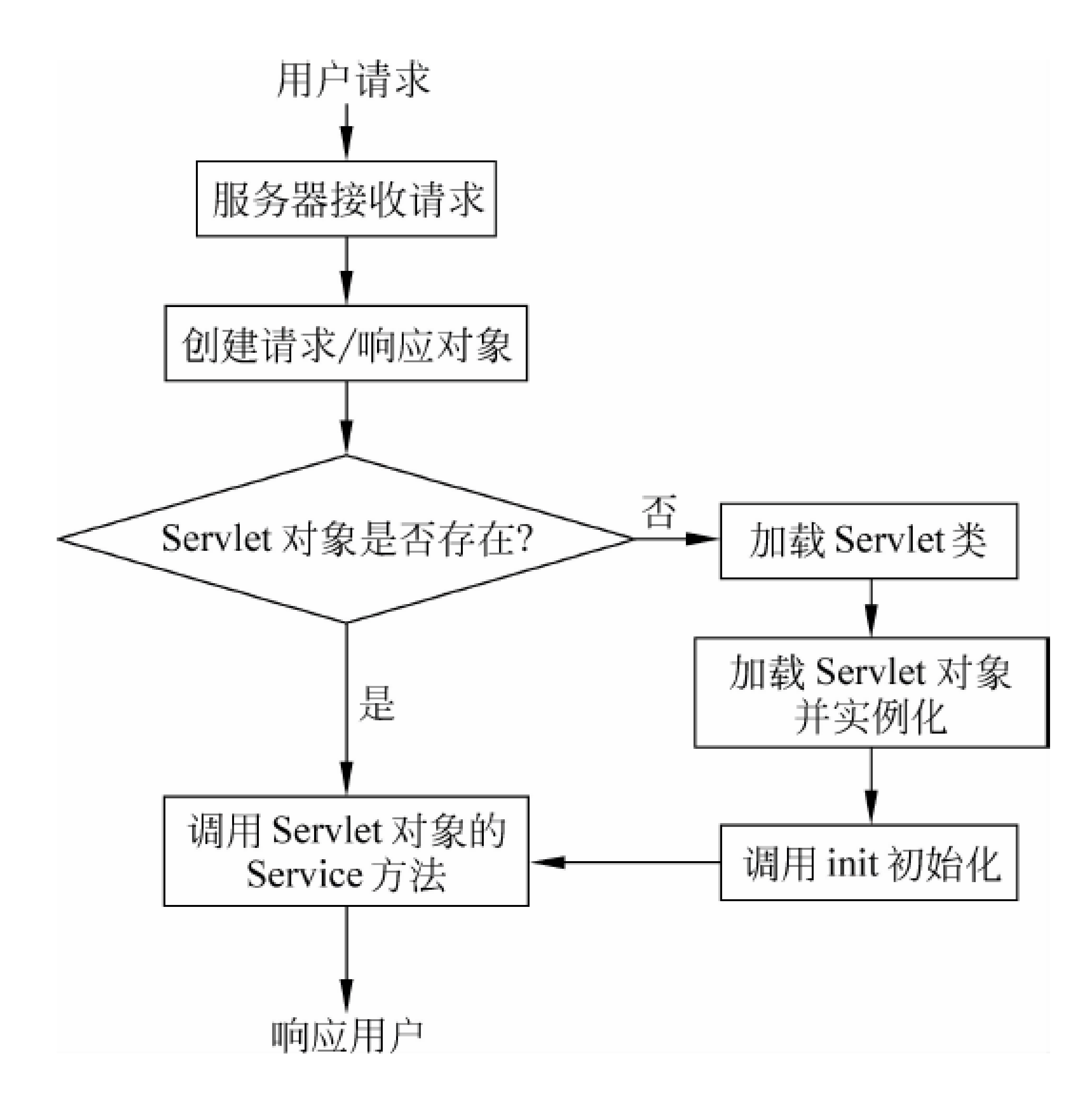


图 7-1 Servlet 的运行原理

如果是启动服务器时加载，请求的第一次访问变慢，但是如果一直没有人访问，则这段时间就浪费了空间。

加载 Servlet 的方式可以在 web.xml 中声明 Servlet 时配置。如果希望在启动时加载 HelloServlet，可使用下面的代码声明：

```
<servlet>
  <servlet-name>hello</servlet-name>
  <servlet-class>servlets.HelloServlet</servlet-class>
   <load-on-startup>1</load-on-startup>
</servlet>
```

7.2 使用 Servlet 完成控制

7.2.1 接收用户的输入信息

在 Servlet 中，doPost 和 doGet 服务方法中都有一个参数是 HttpServletRequest 的对象，用户的请求信息存放在该对象中，要获取用户提交的数据要通过该对象获取。获取用户请求信息的方法有两种：

① String getParameter(String name)
② String[] getParamterValues(String name)

第一种方法用于获取单值表单元素的值，参数指出要获取的表单元素的名字。例如，用户在输入用户编号的时候使用的表单元素的名字是 userid，要想获取用户在这个输入框中输入的值，就可以使用下面的代码：

```
String userid = request.getParameter("userid");
```

如果参数指定的表单元素的名字不存在，则返回 null。所以通常在处理时，要先判断返回的结果是否为 null。

第二种方法用于获取多值表单元素的值，参数也是表单元素的名字，但是返回值是一个数组。

7.2.2 调用业务方法

因为 Servlet 是 Java 类，所以调用业务方法的代码就是在 Java 中学习的调用其他类的方法。首先要创建所使用的 JavaBean 的对象，然后再调用业务方法。

要使用 UserBean 对象，可以使用下面的代码：

```
UserBean user = new UserBean();
```

要调用业务方法，可以使用下面的代码：

```
boolean b = user.validate(userid,userpass);
```

7.2.3 使用 request 存储执行的结果

Servlet 完成控制的功能，查询的结果通过视图(JSP 文件)显示，要把查询的结果传递给 JSP 文件。因为打开这两个文件属于同一次请求，所以共享同一个 request。因为共享同一个 request，所以在 Servlet 中把查询结果存储在 request 中，然后在视图中显示结果。

在登录功能中无须传递信息。假设要把使用 temp 对象保存的查询结果传递到视图，可以使用下面的代码：

```
request.setAttribute("user",temp);
```

7.2.4 选择界面对用户进行响应

Servlet 中的最后一项任务就是跳转到显示界面，在显示界面显示信息。在 JSP 中要重定向到某个文件，可以使用<jsp:forward>标签。那么在 Servlet 中如何完成这种跳转呢？在 Servlet 中提供了一个 RequestDispatcher 接口，通过这个接口可以完成重定向功能。

要使用 RequestDispatcher，需要先创建该类的对象，可以通过 request 的 getRequestDispatcher 方法创建。该方法需要一个 String 类型的参数，该参数指定要转向的文件。假设要转向 userlist.jsp，可以使用下面的代码：

```
RequestDispatcher rd = request.getRequestDispatcher("userlist.jsp");
```

要完成转向，还要调用一个方法，用法如下：

```
rd.forward(request,response);
```

在学习<jsp:forward>标签时还学习过一个标签<jsp:include>。<jsp:include>标签对应的功能在 Servlet 中的实现也是通过上面的 RequestDispatcher 来实现的。例如，现在要包含 userlist.jsp，而不是转向 success.jsp，可以使用下面的代码：

```
RequestDispatcher rd = request.getRequestDispatcher("success.jsp");
rd.include(request,response);
```

实际上，还有一个方法，即 response 对象的 sendRedirect 方法，可以完成重定向的功能。例如，要转向 success.jsp，可以使用下面的代码：

```
response.sendRedirect("success.jsp");
```

但是对于当前的功能来说，不能使用这个方法，因为这个方法与上面介绍的方法有一些区别。在使用 RequestDispatcher 时，当前文件和要转向的文件属于同一次请求，可以共享 request 对象。而在使用 response 对象的 sendRedirect 方法时，当前文件和要转向的文件不属于同一次请求，属于两次请求，所以不能共享 request。如果需要通过 request 对象传递信息，就不能使用 response 的 sendRedirect 方法。如果当前文件和要转向的文件无须共享任何信息，那么这时两种方式都可以。

【例 7-2】 通过界面选择对用户进行响应。

完整的代码如下：

```
package servlets;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import beans.*;
public class LoginServlet extends HttpServlet{
   public void doGet(HttpServletRequest request,HttpServletResponse response)
       throws IOException,ServletException{
//获取用户输入的用户 ID 和口令
       String userid = request.getParameter("userid");
       String userpass = request.getPrameter("userpass");
       //创建模型对象
       UserBean user = new UserBean();
       //调用业务方法进行验证
       boolean b = user.validate(userid,userpass);
       //要转向的文件
       String forward;
//如果登录成功,就把用户名写入 session 中,并且转向 success.jsp,
//否则转向 failure.jsp
       if(b){
//目标转向文件是 success.jsp
          forward = "success.jsp";
       }else{
         //目标转向文件是 failure.jsp
          forward = "failure.jsp";
       }
       //获取 Dispatcher 对象
       RequestDispatcher dispatcher = request.getRequestDispatcher(forward);
       //完成跳转
       dispatcher.forward(request,response);
     }
  }
```

LoginServlet 编写完之后需要配置。在 web.xml 中配置如下：

```
<servlet>
  <servlet-name>login</servlet-name>
  <servlet-class>servlets.LoginServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>login</servlet-name>
  <url-pattern>/login</url-pattern>
</servlet-mapping>
```

要在 login.jsp 中访问 Servlet，要修改 form 表单中的 action 属性，修改如下：

```
<form name="form1" action="login" method="post"
onsubmit="return isValidate(form1)">
```

7.3 JavaBean 简介

JavaBean 是使用 Java 语言编写的组件。该组件的特点是：能够完成特定的功能，这些功能主要用于共享；该组件是一个大的系统的组成部分，不能独立运行；该组件的主要目的是供其他文件进行调用。

JavaBeans 其实是一个 Java 类，只是对类的定义进行了一定的限制，限制的目的是使一个 JavaBeans 具有通用性，并且能被 JavaBeans 的容器所识别，这样用户只要遵守 JavaBeans 的使用要求，不用编写代码就可以在任何一个程序中使用 JavaBeans。

一个 JavaBeans 组件由属性、方法和事件三部分组成。

1. JavaBeans 的属性

JavaBeans 的属性用来描述组件的内部状态。一般对组件的操作都会涉及属性，或者说对组件所有操作的最终目的就是控制它们的属性，通过属性的改变来改变内部状态。

属性的声明与一个 Java 类中一般属性的声明类似。一个属性包括属性名和属性的类型。

JavaBeans 的属性类型可以是 Java 语言中的基本数据类型，也可以是复合数据类型（如类）。

2. JavaBeans 的方法

要对 JavaBeans 的属性进行操作或要实现其他的运算和处理功能，就必须在 JavaBeans 中定义方法。

JavaBeans 的方法表示了一个 JavaBeans 对外提供的功能和服务。

3. JavaBeans 的事件

JavaBeans 的事件与 Java 图形用户界面中的事件模型类似，用于响应用户的操作。

7.3.1 编写 JavaBean

根据以上特点，要编写一个 JavaBean，实际上就是编写一个 Java 类，即编写私有属性、对私有属性进行操作的方法、无参数构造方法和业务方法。

下面是一个对用户信息进行验证的 JavaBean。用户信息非常多，为了简化代码，在 JavaBean 中仅仅写了两个属性：username 和 userpass。

```
package bean;
    public class UserBean
    {
        private String username;
        private String userpass;
        public String getUsername()
        {
            return username;
        }
        public void setUsername(String username)
        {
            this.username = username;
        }
        public String getUserpass()
        {
        return userpass;
                }
                public void setUserpass(String userpass)
                {
                  this.userpass = userpass;
                }
                public int check()
                {
                    if(username.equals("zhangsan") && userpass.equals("wangwu"))
                        return 0;
                    else if(username.equals("wangwu") && userpass.equals("zhangsan"))
                        return 1;
                    else
                        return - 1;
                }
            }
```

这段代码中定义了两个 String 类型的成员变量 username 和 userpass，它们的访问控制类型都是 private。这段代码提供了两个对成员变量 username 进行操作的方法：一个是赋值的方法 setUsername，参数是要赋的值，没有返回值，所以返回值类型是 void；另一个是获取值的方法 getUsername，返回值类型和成员变量的类型一致，没有参数。这段代码同样也提供了两个对 userpass 属性进行操作的方法。

另外，这段代码还定义了一个 check 方法，这个方法的作用是验证用户名和口令是否正确，以及相应的权限。如果返回值是－1，则表示用户登录失败；如果返回值是 0，则表示当前用户是普通用户；如果返回值是 1，则表示当前用户是管理员。这个方法就是这个 JavaBean 的业务方法。

因为这个类不在分布式环境下使用，所以可以不实现 java.io.Serializable 接口。另外，在这个类中没有提供无参数的构造方法，但实际上使用的是系统提供的默认的无参数构造方法。

7.3.2 编译 JavaBean

选择"开始"→"运行"命令，通过命令行进入到 Java 文件所在的位置，然后输入下面的命令进行编译：

```
javac UserBean.java
```

如果没有语法错误，则编译通过。图 7-2 显示了这个过程。

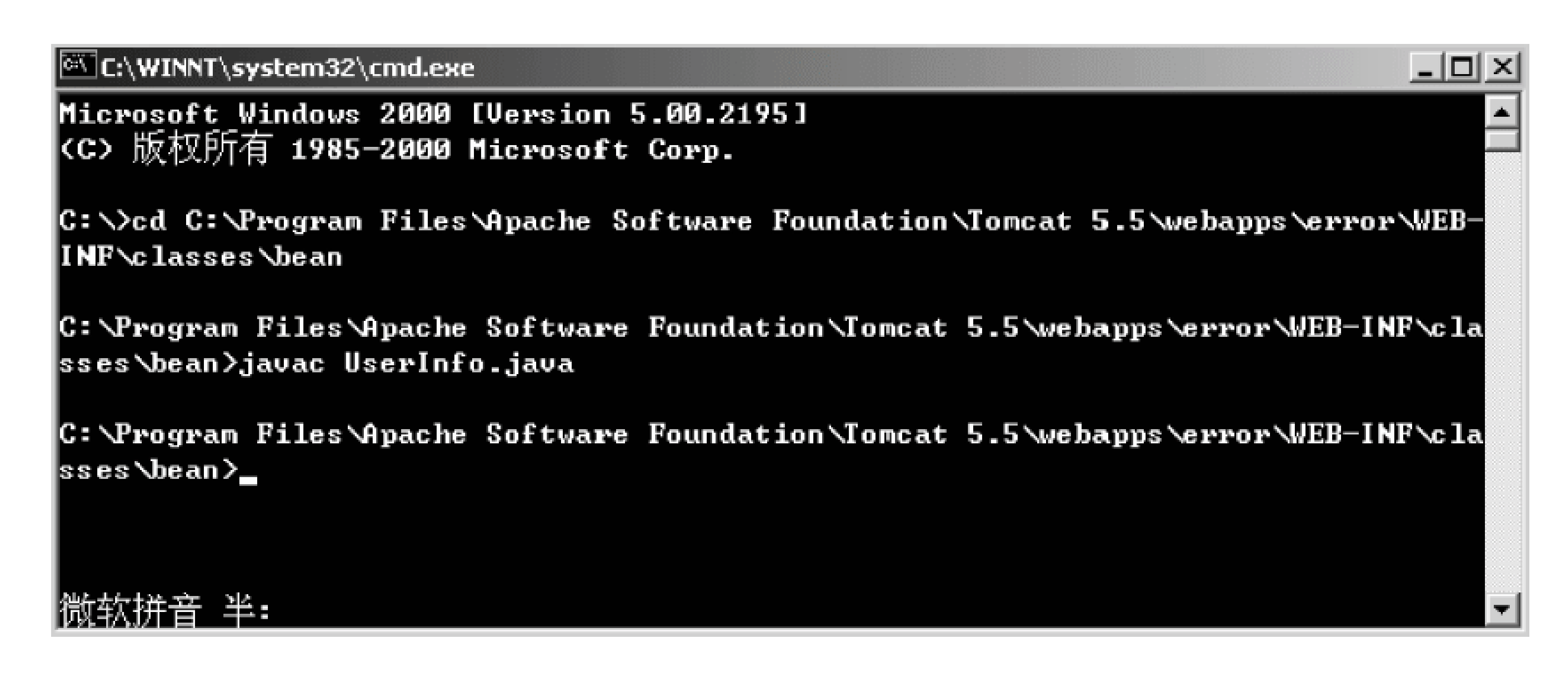


图 7-2 JavaBean 编译过程

7.4 部署 JavaBean

JavaBean 编写完之后，要部署到相应的位置才可以运行。可以把 JavaBean 放在很多位置，只要 Web 应用在运行时能够访问到即可。实际上所有的 Java 文件的部署位置是相同的，包括我们自己编写的 JavaBean，也包括已经存在的 JavaBean。在实际应用中使用已经存在的 JavaBean 的机会更多。

在 Web 应用中，如果这些 Java 文件仅仅是与当前这个 Web 应用相关的，则可以放在这个 Web 应用中；如果这些 Java 文件是同一个服务器上的所有应用共享的，则应该把 Java 文件放在所有 Web 应用都能够访问的地方，而不是某个具体的 Web 应用中。

对于某个特定的 Web 应用使用的 Java 文件，应该放在 WEB-INF 目录下的 classes 子目录中。如果 Java 文件是以压缩包的形式出现的，则应该放在 WEB-INF 目录下的 lib 子目录中。所有 Java 文件的处理方式是相同的。

对于共享的 Java 文件，这些文件一般都以压缩包的形式出现，可以放在服务器的类路径中，通常把压缩包放在 tomcat 安装目录下的 lib 子目录下。放在这个位置的 Java 文件，在所有 Web 应用中都可以被访问。还可以把类文件放在 Java 的类路径上，例如放在 C:\Program Files\Java\jdk1.5.0_01\jre\lib\ext 下。

把 JavaBean 放在 WEB-INF\classes 下面，需要在应用中的 WEB-INF 目录下创建 classess 子目录，然后把编译好的文件放到 classes 目录中。如果 Java 文件中定义了包，则应该在 classes 中创建包的信息。例如，在图 7-2 中的 UserBean 中定义了包名"bean"，所以编译好的文件 UserBean.class 应该位于 WEB-INF\classes\bean 中。

7.4.1　使用 JavaBean

使用 JavaBean 完成口令验证，基本过程如下：

首先声明 JavaBean 的对象，并使用无参数构造方法进行实例化：

```
UserBean user = new UserBean();
```

接着调用 setUsername 和 setUserpass 方法对两个属性赋值：

```
user.setUsername(username);
user.setUserpass(userpass);
```

然后调用业务方法：

```
int result = user.check();
```

进而可以根据该方法返回的结果进行相应的处理。

如果把这样的代码直接放在 JSP 文件中也可以，这样就是 JSP 的最初形式。在 HTML 代码中嵌入 Java 代码，写起来非常方便，但是因为 Java 代码和 HTML 代码夹杂在一起维护起来很不方便，所以在 JSP 中提供了几个对 JavaBean 进行操作的标签，使用这些标签可以避免使用 Java 代码。标签的使用非常简单，与普通的 HTML 标签的使用非常类似。

7.4.2　JavaBean 标签使用

1. <jsp:useBean>标签

<jsp:useBean>标签的作用就是声明一个对象，如果这个对象已经存在，就查找到这个对象；如果这个对象不存在，就创建对象。它与代码 UserBean user = new UserBean()的作用类似。该标签的基本语法格式如下：

```
<jsp:useBean id="对象名" class="完整的类名" scope="作用范围"/>
```

id 属性指出定义的对象的名字，这个名字应该符合 Java 变量命名规范。class 属性指出这个变量的类型，应该是类的完整类型，包含类的包的信息。scope 属性指出这个变量的作用范围，有 4 种可能的选项：page、request、session 和 application。通常我们定义的变量仅仅是在本次请求内使用，所以通常使用 request。如果仅仅是在当前页面使用，可以使用 page。如果要共享，可以使用 session 或者 application。

这个标签会使用 class 指定的类创建一个对象，对象的名字是属性 id 指定的名字，对象的作用范围是 scope 属性指出的作用范围。

要想创建 UserBean 的对象，需要知道完整的类名、对象的名字和对象的作用范围。这里类的名字是 bean.UserBean，对象名字是 user，作用范围是本次请求，所以使用<jsp:useBean>改写后的代码如下：

```
<jsp:useBean id="user" class="bean.UserBean" scope="request"/>
```

2. <jsp:setProperty>标签

<jsp:setProperty>标签用于对 JavaBean 的属性赋值，可以对某个 JavaBean 的某个属

性赋值，也可以对 JavaBean 的所有属性赋值。它与直接调用 setUsername 和 setUserpass 的作用是相同的。

<jsp:setProperty>标签的基本格式如下：

```
<jsp:setProperty name = "JavaBean 的名字" property = "属性的名字"
  [value = "值"] [param = "表单元素名字"]/>
```

要为某个 JavaBean 的某个属性赋值，需要知道为哪个 JavaBean 的哪个属性赋值，以及赋什么样的值。name 属性指出要对哪个 JavaBean 赋值，该 JavaBean 应该是已经存在的 JavaBean。property 属性指出要为 JavaBean 的哪个属性赋值，应该是 JavaBean 的某个属性的名字。value 或者 param 指出要赋什么样的值，value 直接指出值，param 指出使用用户的表单请求数据中表单元素的名字。

要为前面声明的 JavaBean 对象 user 的 username 属性赋值 lixucheng，可以使用下面的代码：

```
<jsp:setProperty name = "user" property = "username" value = "lixucheng"/>
```

value 的值可以使用表达式语言，这样就可以使用前面定义好的信息。假设在 request 中已经存在保存用户名的 username 变量，可以写成下面的样子：

```
<jsp:setProperty name = "user" property = "username" value = "${username}"/>
```

如果这个值是来自请求信息中的表单元素 username，则可以写成下面的样子：

```
<jsp:setProperty name = "user" property = "username" value = "${param.username}"/>
```

如果要赋的值来源于请求表单元素，还可以写成下面的样子：

```
<jsp:setProperty name = "user" property = "username" param = "username"/>
```

如果表单元素的名字和要赋值的属性的名字相同，则可以省略 param 属性(效果是相同的)：

```
<jsp:setProperty name = "user" property = "username"/>
```

也就是说，如果没有给定值，系统会查看请求表单元素，看有没有与当前要赋值的属性的名字相同的，如果有，则获取这个表单元素的值，然后给 JavaBean 的属性赋值。

要想为 JavaBean 的 userpass 属性赋值，可以使用下面的代码：

```
<jsp:setProperty name = "user" property = "userpass"/>
```

这时分别从表单元素中取出用户名和口令，然后分别对 Java 的属性 username 和 userpass 进行赋值。在这样的情况下，两个赋值语句可以写在一起，如下所示：

```
<jsp:setProperty name = "user" property = "*"/>
```

该语句表示为 user 对象的所有属性赋值。

综上所述，<jsp:setProperty>标签存在以下 4 种形式：

- <jsp:setProperty name ＝"JavaBean 的名字" property＝"属性的名字" value＝"值"/>。

- <jsp:setProperty name="JavaBean 的名字" property="属性的名字" param="表单元的名字"/>。
- <jsp:setProperty name="JavaBean 的名字" property="属性的名字"/>。
- <jsp:setProperty name="JavaBean 的名字" property=" * "/>。

3. 使用<jsp:getProperty>

<jsp:getProperty>标签用于获取 JavaBean 的属性的值,并显示在相应的位置上,其作用与调用 getUsername 和 getUserpass 的作用是相同的。

<jsp:getProperty>标签的基本格式如下:

```
<jsp:getProperty name = "JavaBean 的名字" property = "属性的名字"/>
```

name 属性指定要获取哪个 JavaBean 的属性,这个 JavaBean 应该是一个已经存在的 JavaBean。property 属性指定获取 JavaBean 的哪个属性,是 JavaBean 的属性的名字。要想获取属性 username 的值,可以使用下面的代码:

```
<jsp:getProperty name = "user" property = "username"/>
```

这个代码与下面代码的作用完全相同:

```
${user.username}
```

7.5 本章小结

Servlet 是一种 Web 组件,与 JSP 完成的功能完全相同。JSP 的优势在于开发界面,而 Servlet 的优势在于控制。

Servlet 是一个 Java 类,但它与 JavaBean 不同。Servlet 可以接收用户的请求,可以对用户进行响应,而 JavaBean 只能被其他的文件调用。

编写在 Web 上使用的 Servlet 都要继承一个 HTTPServlet 类。Servlet 中主要包括 3 类方法:init 初始化方法、doGet 和 doPost 等 service 方法以及 destroy 方法。在 Servlet 的生命周期中,每次请求都会调用 service 方法,而只有在创建 Servlet 实例之后才会调用 init 方法进行初始化,且只执行一次。

Servlet 编译之后需要放到 WEB-INF 目录下的 classes 子目录中,并且需要在 web.xml 文件中进行配置。

在 Servlet 中可以通过第一个参数 request 的 getParameter 方法和 getParameterValues 方法获取用户输入的信息,可以通过 response 对象获取输出流对象 out 来对用户进行响应。

Servlet 作为控制器主要完成以下工作:

(1) 获取用户提交的信息。

(2) 调用业务方法。

(3) 通过向 request 中保存信息来向页面传递信息。

(4) 根据业务方法的返回结果对用户进行响应。

JavaBean 是组件,但是不能独立运行。JavaBean 能够封装一些功能,并且这些功能是

一些通用的功能，主要供其他应用来调用。

JavaBean 就是普通的 Java 类，在编写 JavaBean 时要提供无参数构造方法，要编写私有属性以及对属性进行操作的公有方法，要封装常用的功能。

JavaBean 编写之后要放在 Web 应用的 WEB-INF 目录下的 classes 子目录中。要按照包名组织文件。

7.6 思考与练习

1. JSP 和 Servlet 有什么相同点？有什么不同点？

2. Servlet 和 JavaBean 有什么相同点？有什么不同点？

3. 简述 Servlet 的生命周期。

4. 是不是对 Servlet 的每个请求都会创建一个 Servlet 对象？如果不是，那么 Servlet 在响应每个用户的请求时有什么区别？

5. 编写好的 Servlet 需要放在什么地方？

6. 如何理解 JavaBean？

7. JavaBean 与 JSP 的区别是什么？

8. 在 JavaBean 中有一个属性 usertype，类型是 String，现在希望添加两个对属性进行操作的 setter 方法和 getter 方法。请写出这两个方法。

9. 有一个 JavaBean，类名是 DoAll，包名是 com. nf. ma，请写出这个 JavaBean 在 Web 应用中存放的位置。

10. 使用<jsp:useBean>标签声明的 JavaBean 对象的作用范围有哪些？这些作用范围有什么区别？

11. 有一个 JavaBean，类名是 MyBean，包名是 mypackage，现在希望使用这个类创建一个名字为 bean1 的对象，这个对象的作用范围是 session。请使用<jsp:useBean>标签完成这个功能。

12. 在表单中有一个输入框，名字为 myinput，现在希望把用户输入的信息赋值给第 11 题中定义的 bean1 对象的属性 myProperty。请使用<jsp:setProperty>标签完成该功能。

13. 现在希望把第 11 题的 bean1 对象的 myProperty 属性输出到界面上，请使用<jsp:getProperty>标签完成。

14. 在 Java 代码中如何使用 JavaBean(包括 JavaBean 对象的创建、对 JavaBean 对象的属性赋值、得到 JavaBean 对象的属性值、调用 JavaBean 对象的业务方法)？

第8章 JSP内置对象

通过本章的学习让学生了解JSP内置对象的基本关系；理解session对象和application对象的方法；掌握request对象获取信息、request对象处理汉字信息、response对象改变HTTP头、response对象重定向、response的状态行、out对象的方法；并能够运用所学实现计数器、留言板。

学习目标及要求

(1) 了解文件类、流的概念。

(2) 理解字节流、字符流、回压字符流、数据流、对象流、RandomAccessFile流。

(3) 掌握以上各种流的文件操作方法。

(4) 应用文件上传、文件下载。

有些对象不用声明就可以在JSP页面的脚本部分使用，这就是JSP的内置对象。

JSP的内置对象有resquest、response、session、application和out。

8.1 request对象

HTTP通信协议是客户与服务器之间一种提交（请求）信息与响应信息（request/respone）的通信协议。在JSP中，内置对象request封装了用户提交的信息，那么该对象调用相应的方法可以获取封装的信息，即使用该对象可以获取用户提交的信息。

客户通常使用HTML表单向服务器的某个JSP页面提交信息，表单的一般格式如下：

```
<FORM  method = get | post  action = "提交信息的目的地页面">
    提交手段
</FORM>...
```

其中，<FORM>是表单标签，method取值get或post。get方法和post方法的主要区别是：使用get方法提交的信息会在提交的过程中显示在浏览器的地址栏中；而使用post方法提交的信息不会显示在地址栏中。提交手段包括通过文本框、列表、文本区等。例如：

```
<FORM action = "tom.jsp" method =  "post" >
     < INPUT type = "text" name = "boy" value =  "ok" >
     < INPUT type = "submit" value = "送出" name =  "submit">
</FORM>
```

该表单使用 post 方法向页面 tom.jsp 提交信息，提交信息的手段是在文本框输入信息，其中默认信息是“ok”，然后单击“送出”按钮向服务器的 JSP 页面 tom.jsp 提交信息。

request 对象可以使用 getParameter(String s)方法获取该表单通过 text 提交的信息，例如 request.getParameter("boy")。

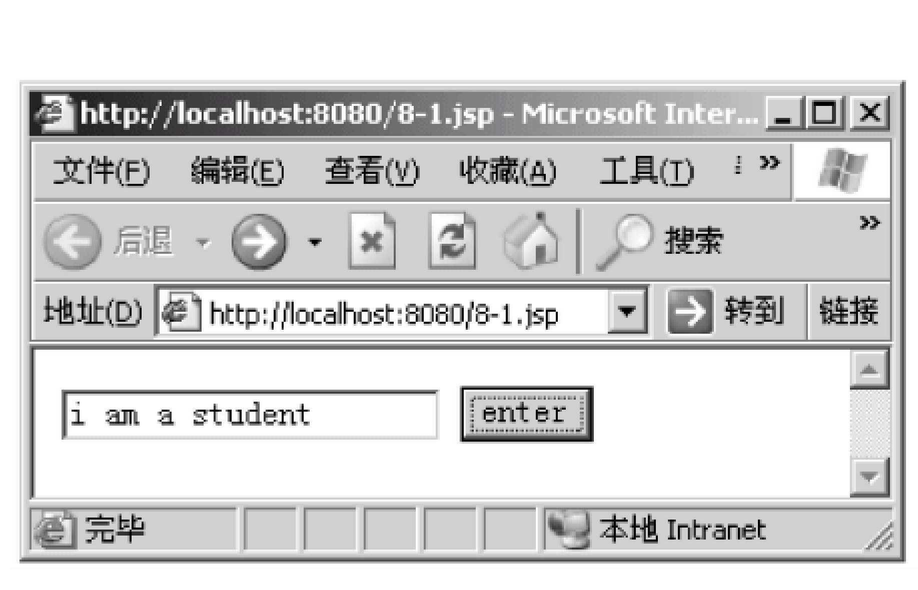
8.1.1 获取客户提交的信息

request 对象获取客户提交信息的最常用的方法是 getParameter(String s)。在下面的例 8-1 中，Example8_1.jsp 通过表单向 tree.jsp 提交信息“I am a student”；tree.jsp 通过 request 对象获取表单提交的信息，包括 text 的值以及按钮的值。

在本章中，例子中所涉及的 JSP 页面都保存在 Web 服务目录的根目录 Root 中，即 D:tomcat/jakarta-tomcat-4.0/webapps/Root 中。

【例 8-1】 通过表单向 tree.jsp 提交信息(如图 8-1 所示)。

```
Example8_1.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY bgcolor = green><FONT size = 1>
   <FORM action = "tree.jsp" method = post name = form>
        <INPUT type = "text" name = "boy">
        <INPUT type = "submit" value = "Enter" name = "submit">
   </FORM>
</FONT>
</BODY>
</HTML>
tree.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY bgcolor = green><FONT size = 1>
<P>获取文本框提交的信息:
   <% String textContent = request.getParameter("boy");   %>
<BR><% = textContent %>
<P> 获取按钮的名字:
   <% String buttonName = request.getParameter("submit");   %>
<BR><% = buttonName %></FONT></BODY></HTML>
```



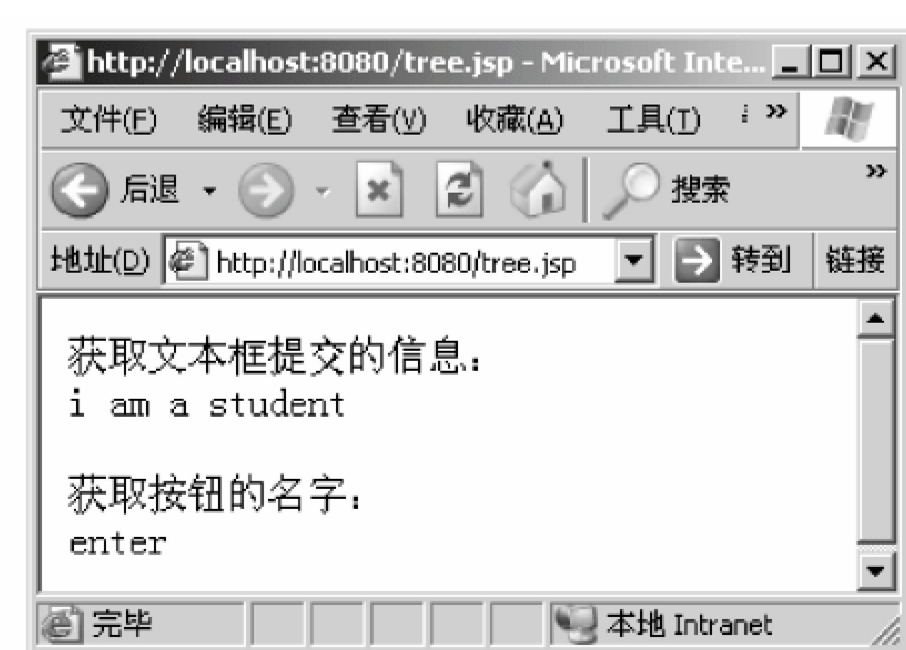


图 8-1 提交信息与获取信息

在下面的例 8-2 中,Example8_2.jsp 通过表单向自己提交一个正数,然后计算这个数的平方根。

【例 8-2】 通过表单向自己提交一个正数,然后计算这个数的平方根(如图 8-2 所示)。

```
Example8_2.jsp:
<%@ page contentType="text/html;charset=GB2312" %>
<HTML>
<BODY bgcolor=cyan><FONT size=1>
  <FORM  action="Example8_2.jsp" method=post name=form>
        <INPUT type="text" name="girl">
        <INPUT TYPE="submit" value="Enter" name="submit">
  </FORM>
  <% String textContent=request.getParameter("girl");
    double number=0,r=0;
    if(textContent==null)
        {textContent="";          }
    try{ number=Double.parseDouble(textContent);
          if(number>=0)
             {r=Math.sqrt(number) ;
              out.print("<BR>"+String.valueOf(number)+"的平方根：");
              out.print("<BR>"+String.valueOf(r)); }
          else
             {out.print("<BR>"+"请输入一个正数"); }}
    catch(NumberFormatException e)
        {out.print("<BR>"+"请输入数字字符");} %>
</FONT></BODY></HTML>
```

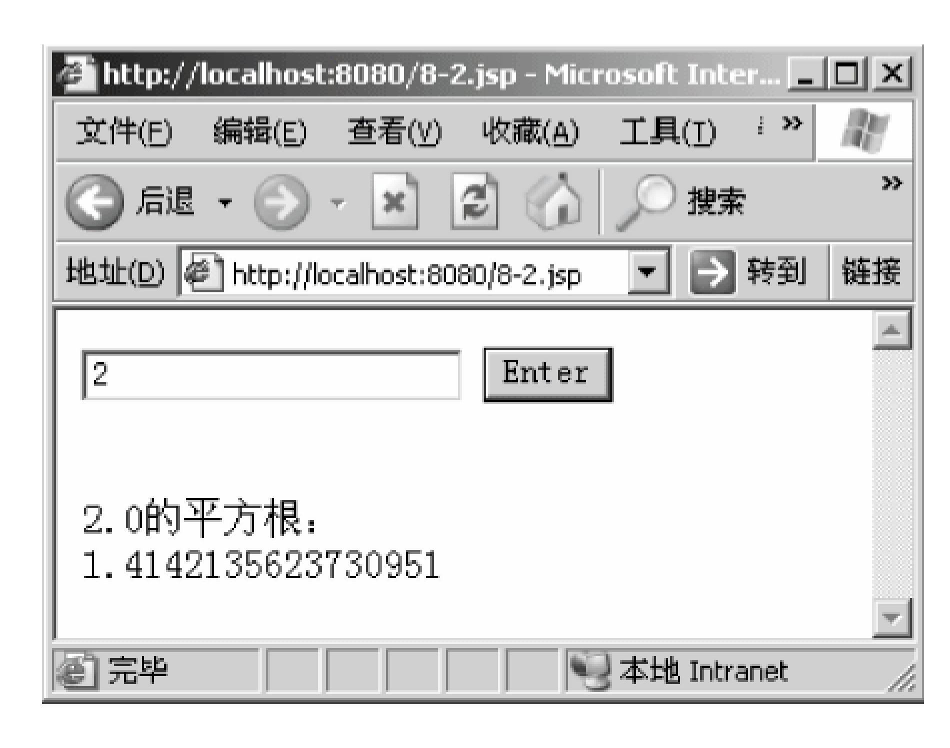


图 8-2 计算数的平方根

8.1.2 处理汉字信息

当用 request 对象获取客户提交的汉字字符时会出现乱码问题,所以对含有汉字字符的信息必须进行特殊的处理。首先,将获取的字符串用 ISO-8859-1 进行编码,并将编码存放到一个字节数组中,然后将这个数组转化为字符串对象即可。如下所示:

```
String str=request.getParameter("girl");
byte  b[ ]=str.getBytes("ISO-8859-1");
str=new String(b);
```

通过上述过程，提交的任何信息（无论是中文字符或西文字符）都能正确地显示。

下面的例 8-3 对例 8-1 按上述办法做了改动，并将按钮上的字变成汉语，在文本框里输入“苹果：apple:12 斤 5 $”，然后提交给 tree.jsp。

【例 8-3】 将按钮上的字变成汉语，在文本框里输入“苹果：apple:12 斤 5 $”，然后提交给 tree.jsp（如图 8-3 所示）。

```
Example8_3.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML>
<BODY bgcolor = green><FONT size = 1>
   <FORM action = "tree.jsp" method = post name = form>
        <INPUT type = "text" name = "boy">
        <INPUT type = "submit" value = "提交" name = "submit">
   </FORM></FONT></BODY></HTML>
tree.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<MHML><BODY>
<P>获取文本框提交的信息:
   <% String textContent = request.getParameter("boy");
     byte  b[] = textContent.getBytes("ISO - 8859 - 1");
     textContent = new String(b); %>
<BR>  <% = textContent %>
<P> 获取按钮的名字:
   <% String buttonName = request.getParameter("submit");
     byte  c[] = buttonName.getBytes("ISO - 8859 - 1");
     buttonName = new String(c); %>
<BR>   <% = buttonName %></BODY></HTML>
```

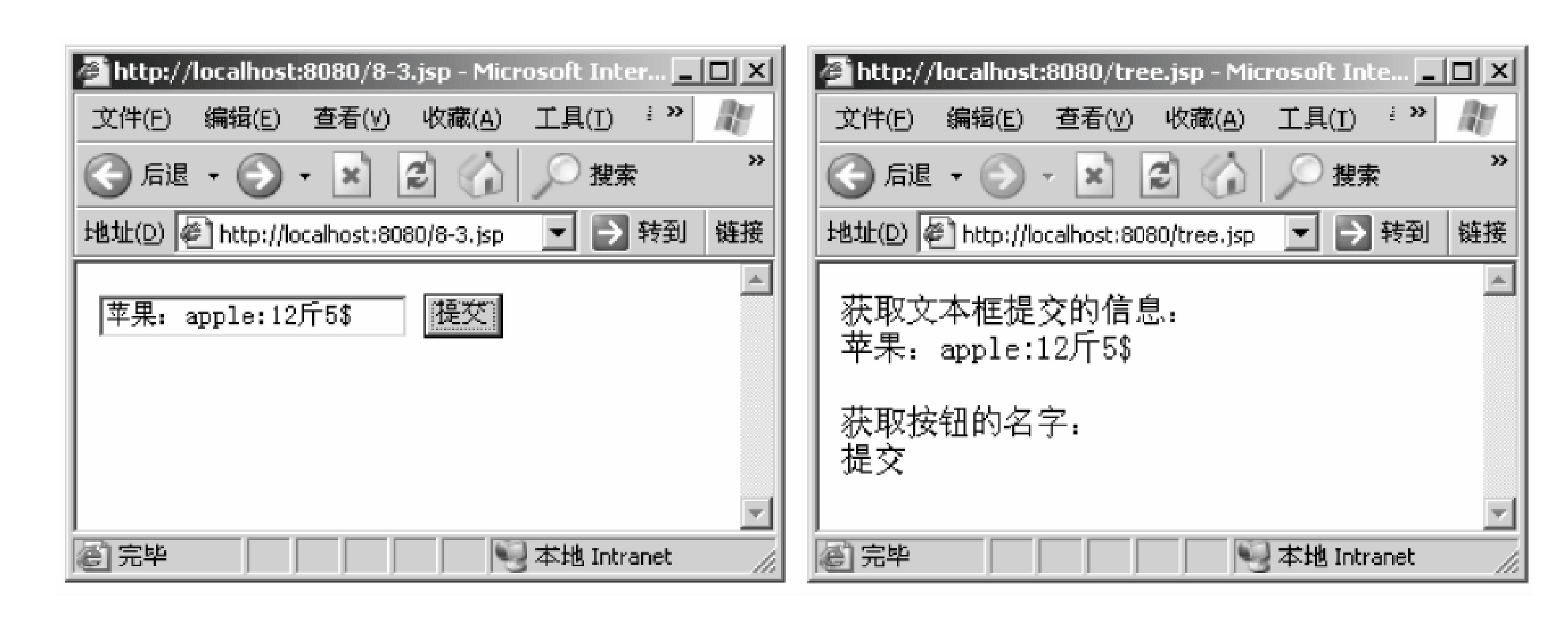


图 8-3 提交和获取含有汉字的信息

8.1.3 常用方法举例

当客户访问一个页面时，会提交一个 HTTP 请求给服务器的 JSP 引擎，这个请求包括一个请求行、http 头和信息体，如下所示：

```
post/tree2.jsp/HTTP.1.1
host: localhost: 8080
```

```
accept - encoding: gzip, deflate
```

其中，首行叫请求行，规定了向访问的页面请求提交信息的方式，如 post、get 等，以及请求的页面的文件名字和使用的通信协议。

第二、三行分别是两个头（Header），称 host、accept-encoding 是头名字，而 localhost:8080 以及 gzip、deflate 分别是它们的值。这里 host 的值是 tree2.jsp 的地址。上面的请求有两个头：host 和 accept-encoding，一个典型的请求通常包含很多的头，有些头是标准的，有些头和特定的浏览器有关。

一个请求还包含信息体，即 HTML 标记组成的部分，可能包括各式各样用于提交信息的表单等，如：

```
<BODY>
  <FORM action = "tree2.jsp" method = post name = form>
      <INPUT type = "text" name = "boy">
      <INPUT type = "submit" value = "" name = "submit">
  </FORM>
</BODY>
```

可以使用 JSP 引擎的内置对象 request 来获取客户提交的信息，说明如下。

(1) getProtocol()：获取客户向服务器提交信息所使用的通信协议，如 http/1.1 等。

(2) getServletPath()：获取客户请求的 JSP 页面文件的目录。

(3) getContentLength()：获取客户提交的整个信息的长度。

(4) getMethod()：获取客户提交信息的方式，如 post 或 get。

(5) getHeader(String s)：获取 HTTP 头文件中由参数 s 指定的头名字的值，一般来说，s 参数可取的头名有 accept、referer、accept-language、content-type、accept-encoding、user-agent、host、content-length、connection、cookie 等。例如，s 取值 user-agent 将获取客户的浏览器的版本号等信息。

(6) getHeaderNames()：获取头名字的一个枚举。

(7) getHeaders(String s)：获取头文件中指定头名字的全部值的一个枚举。

(8) getRemoteAddr()：获取客户的 IP 地址。

(9) getRemoteHost()：获取客户机的名称(如果获取不到，就获取 IP 地址)。

(10) getServerName()：获取服务器的名称。

(11) getServerPort()：获取服务器的端口号。

(12) getParameterNames()：获取客户提交的信息体部分中 name 参数值的一个枚举。

下面的例 8-4 使用了 request 的一些常用方法。

【例 8-4】　request 的常用方法(如图 8-4、图 8-5 所示)。

```
Example8_4.jsp:
<HTML>
<BODY bgcolor = cyan><FONT size = 1>
<%@ page contentType = "text/html;charset = GB2312" %>
   <FORM action = "tree2.jsp" method = post name = form>
       <INPUT type = "text" name = "boy">
       <INPUT type = "submit" value = "enter" name = "submit">
   </FORM></FONT></BODY></HTML>
```



图 8-4 提交信息

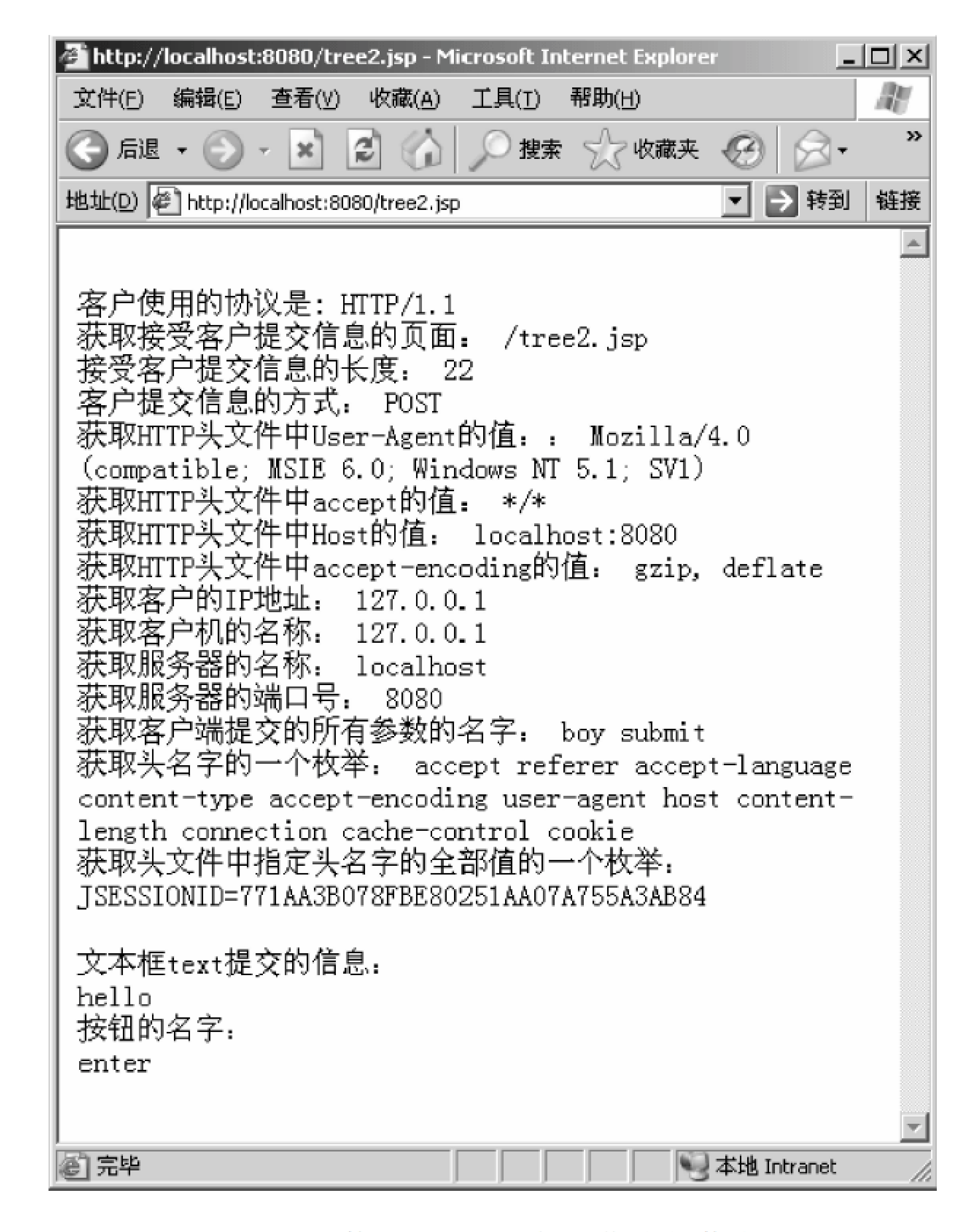


图 8-5 使用 request 方法获取的信息

```
tree2.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<%@ page import = "java.util.*" %>
<MHML>
<BODY bgcolor = cyan>
<Font size = 1 >
<BR>客户使用的协议是:
   <% String protocol = request.getProtocol();
      out.println(protocol);   %>
<BR>获取接受客户提交信息的页面:
   <% String path = request.getServletPath();
      out.println(path);   %>
```

```
<BR>接受客户提交信息的长度:
  <% int length = request.getContentLength();
     out.println(length);   %>
<BR>客户提交信息的方式:
  <% String method = request.getMethod();
     out.println(method);   %>
<BR>获取 HTTP 头文件中 User - Agent 的值: :
  <% String header1 = request.getHeader("User - Agent");
     out.println(header1);   %>
<BR>获取 HTTP 头文件中 accept 的值:
  <% String header2 = request.getHeader("accept");
     out.println(header2);   %>
<BR>获取 HTTP 头文件中 Host 的值:
  <% String header3 = request.getHeader("Host");
     out.println(header3);   %>
<BR>获取 HTTP 头文件中 accept - encoding 的值:
  <% String header4 = request.getHeader("accept - encoding");
     out.println(header4);   %>
<BR>获取客户的 IP 地址:
  <% String  IP = request.getRemoteAddr();
     out.println(IP);   %>
<BR>获取客户机的名称:
  <% String clientName = request.getRemoteHost();
     out.println(clientName);   %>
<BR>获取服务器的名称:
  <% String serverName = request.getServerName();
     out.println(serverName);   %>
<BR>获取服务器的端口号:
  <% int serverPort = request.getServerPort();
     out.println(serverPort);   %>
<BR>获取客户端提交的所有参数的名字:
  <% Enumeration enum = request.getParameterNames();
      while(enum.hasMoreElements())
            {String s = (String)enum.nextElement();
             out.println(s); } %>
<BR>获取头名字的一个枚举:
  <% Enumeration enum_headed = request.getHeaderNames();
     while(enum_headed.hasMoreElements())
            {String s = (String)enum_headed.nextElement();
             out.println(s); }   %>
<BR>获取头文件中指定头名字的全部值的一个枚举:
  <% Enumeration enum_headedValues = request.getHeaders("cookie");
     while(enum_headedValues.hasMoreElements())
            {String s = (String)enum_headedValues.nextElement();
             out.println(s);   }    %><BR>
  <P> 文本框 text 提交的信息:
  <% String str = request.getParameter("boy");
    byte  b[] = str.getBytes("ISO - 8859 - 1");
    str = new String(b);  %><BR><% = str %>
<BR> 按钮的名字:
  <% String buttonName = request.getParameter("submit");
    byte  c[] = buttonName.getBytes("ISO - 8859 - 1");
    buttonName = new String(c);  %>
<BR>  <% = buttonName %></Font></BODY></HTML>
```

8.1.4 用户注册

在下面的例 8-5 中,用户通过提交姓名和 E-mail 地址实现注册。当 request 对象获取这些信息后,首先检查散列表对象中是否已经存在这个名字,该散列表存储了已经注册的用户的名字。如果目前准备注册的用户提交的名字在散列表中已经存在,就提示客户更换名字,否则将检查客户是否提供了书写正确的 E-mail 地址,如果提供了书写正确的 E-mail 地址将允许注册。

我们将例 8-5 中的 login1. jsp 和 login2. jsp 保存在 Web 服务目录 root 中,用户首先访问 login1. jsp,输入名字和 E-mail 并提交给 login2. jsp 实现注册。

【例 8-5】 实现注册功能(如图 8-6 所示)。

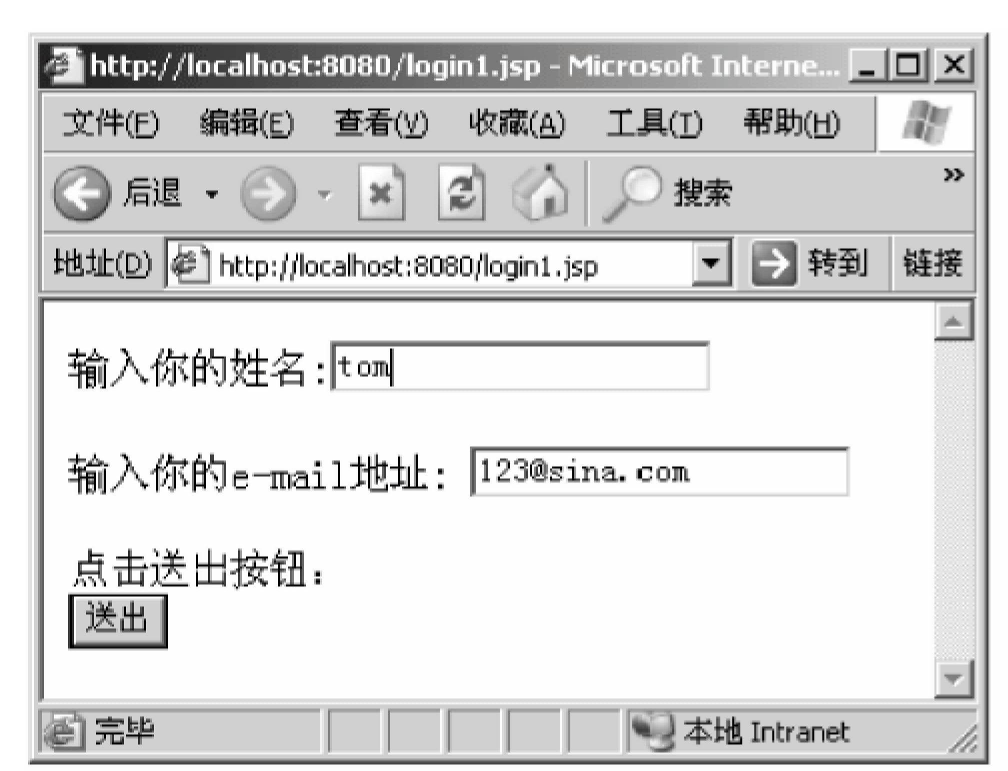




图 8-6 用户注册

```
Login1. jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML><BODY bgcolor = cyan><Font size = 1 >
  <FORM action = "login2. jsp" method = post >
     <P>输入你的姓名:<INPUT type = "text" name = "name" value = "abc">
     <BR><P>输入你的 E-mail 地址:
       <INPUT type = "text" name = "address" value = "ookk@sina.com">
     <P>单击“送出”按钮:<BR>
     <INPUT type = "submit" value = "送出" name = submit >
  </FORM></FONT></BODY></HTML>
login2. jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<%@ page import = "java. util. * " %>
<HTML>
<BODY bgcolor = cyan><Font size = 1 >
  <%!Hashtable hashtable = new Hashtable();
     public synchronized void putString(String s)
     { hashtable. put(s,s);  } %>
  <%  String person_name = request. getParameter("name"),
            name_found = null;
         if(person_name == null)
             {person_name = ""; }
```

```
        byte b[ ] = person_name.getBytes("ISO-8859-1");
                  person_name = new String(b);
        name_found = (String)hashtable.get(person_name);
       if(name_found == null)
          { String person_email = request.getParameter("address");
              if(person_email == null)
                 {person_email = ""; }
            StringTokenizer fenxi = new StringTokenizer(person_email," @");
            int n = fenxi.countTokens();
             if(n >= 3)
                {out.print("<BR>" + "你输入的 E-mail 有不合法字符");}
             else
               { putString(person_name);
                 out.print("<BR>" + "你已经注册成功");
                 out.print("<BR>" + "你注册的名字是" + person_name); } }
             else
          {out.print("<BR>" + "该名字已经存在,请你换个名字");} %>
</FONT></BODY></HTML>
```

8.1.5 获取 HTML 表单提交的数据

一个表单的数据提交手段部分通常包括如下的标记符号：

- <INPUT ...>。
- <Select ... ></Select>。
- <Option ...></Option>。
- <TextArea ...></TextArea>。

1. <INPUT>的基本格式

在表单中用 INPUT 标记来指定表单中数据的输入方式以及表单的提交键。INPUT 标记中的 type 属性可以指定输入方式的 GUI 对象，name 属性用来指定这个 GUI 对象的名称。基本格式如下：

```
<INPUT type = "输入对象的 GUI 类型" name = "名字">
```

服务器通过属性 name 指定的名字来获取“输入对象的 GUI 类型”中提交的数据。“输入对象的 GUI 类型”可以是 text(文本框)、checkbox(检查框)、submit(提交键)等。

1）文本框：text

当输入对象的 GUI 类型是 text 时，除了用 name 为 text 指定名字外，还可以为 text 指定其他的一些值。例如：

```
<INPUT type = "text" name = "me" value = "hi" size = "12 " align = "left"  maxlength = "30">
```

其中，value 的值是 text 的初始值；size 是 text 对象的长度(单位是字符)；align 是 text 在浏览器窗体中的对齐方式；maxlength 指定 text 可输入字符的最大长度。

2）单选框：radio

当输入对象的 GUI 类型是 radio 时，除了用 name 为 radio 指定名字外，还可以为 radio 指定其他的一些值。例如：

```
<INPUT type="radio"name="rad" value="red" align="top"checked= "java">
```

其中,value 指定 radio 的值;align 是 radio 在浏览器窗体中的对齐方式;如果几个单选键的 name 取值相同,那么同一时刻只能有一个被选中。服务器通过 name 指定的名字来获取被选中的 radio 提交的由 value 指定的值。checked 如果取值是一个非空的字符串,那么该单选框的初始状态就是选中状态。

在下面的例 8-6 中,用单选框来实现 radio 按钮测试。客户在 radio.jsp 页面中选中几个单选框,将选择提交给 answer.jsp 页面。

【例 8-6】 用单选框来实现 radio 按钮测试(如图 8-7 所示)。





图 8-7 获取单选框的值

```
radio.jsp:
<HTML>
<%@ page contentType="text/html;charset=GB2312" %>
<BODY bgcolor=cyan><Font size=1>
<P>诗人李白是中国历史上哪个朝代的人:
  <FORM action="answer.jsp" method=post name=form>
      <INPUT type="radio" name="R" value="a">宋朝
      <INPUT type="radio" name="R" value="b">唐朝
      <INPUT type="radio" name="R" value="c">明朝
      <INPUT type="radio" name="R" value="d" checked="ok">元朝
        <BR>
<P>小说红楼梦的作者是:
        <BR>
      <INPUT type="radio" name="P" value="a">曹雪芹
      <INPUT type="radio" name="P" value="b">罗贯中
      <INPUT type="radio" name="P" value="c">李白
      <INPUT type="radio" name="P" value="d">司马迁
      <BR>
      <INPUT type="submit" value="提交答案" name="submit">
  </FORM></FONT></BODY></HTML>
answer.jsp:
<HTML>
<%@ page contentType="text/html;charset=GB2312" %>
<BODY bgcolor=cyan><Font size=1>
<% int n=0;
    String s1=request.getParameter("R");
    String s2=request.getParameter("P");
```

```
        if(s1 == null)
          {s1 = "";}
        if(s2 == null)
          {s2 = "";}
         if(s1.equals("b"))
          { n++ ;}
         if(s2.equals("a"))
          { n++ ;} %>
<P>你得了<% = n %>分</FONT></BODY></HTML>
```

3）复选框：checkbox

当输入对象的 GUI 类型是 checkbox 时，除了用 name 为 checkbox 指定名字外，还可以为 checkbox 指定其他的一些值。例如：

```
<INPUT type = "checkbox"name = "ch"value = "pink" align =  "top"checked =  "java" >
```

其中，value 指定 checkbox 的值。复选框与单选框的区别就是复选框可以多选。服务器通过 name 指定的名字来获取被选中的 checkbox 值。为了使服务器能获取提交的值，复选框 name 的值应互不相同。checked 取值如果是一个非空的字符串，那么该复选框的初始状态就是选中状态。

4）口令框：password

它是输入口令用的特殊文本框，输入的信息用“*”回显，防止他人偷看口令。例如：

```
<INPUT type = "password"name = "me"size = "12" maxlength = "30">
```

服务器通过 name 指定的字符串获取 password 提交的值。例如，在口令框中输入“bird88_1”，那么 bird88_1 将被提交给服务器。口令框仅仅起着不让别人偷看的作用，不提供保密措施。

5）提交键：submit

为了能把表单的数据提交给服务器，一个表单至少要包含一个提交键。例如：

```
<INPUT type = "submit" name = "me"value = "ok"size = "12 ">
```

单击“提交”按钮后，服务器就可以获取表单提交的各个数据。当然服务器也可以通过 name 指定的名字来获取“提交”按钮提交的由 value 指定的值。

6）重置键：reset

重置键将表单中输入的数据清空，以便重新输入数据。

```
<INPUT type =  "reset" >
```

2. <SELECT>、<OPTION>格式

下拉式列表和滚动列表通过<SELECT>和<OPTION>标记来定义，基本格式如下：

```
<SELECT>  <OPTION>  <OPTION>...</SELECT>
```

1）下拉列表

```
<SELECT name = "shulie" >
          <OPTION value = "cat">你选了小猫
```

```
        <OPTION value = "dog">你选了小狗
        ...
       <OPTION value = "600"> n = 600 </SELECT>
```

服务器通过 name 获取下拉列表中被选中的 option 的值(参数 value 指定的值)。

2) 滚动列表

SELECT 中增加 size 属性的值就变成滚动列表,size 的值是滚动列表的可见行的个数。

```
<SELECT name = "shulie" size = 2 >
        <OPTION value = "1">计算 1 到 n 的连续和
        <OPTION value = "2">计算 1 到 n 的平方和
        <OPTION value = "3">计算 1 到 n 的立方和
</SELECT>
```

服务器通过 name 获取滚动列表中被选中的 option 的值(参数 value 指定的值)。

在下面的例 8-7 中,客户通过滚动列表选择计算求和的方式,通过下拉列表选择计算求和的项数。

【例 8-7】 客户通过滚动列表选择计算求和的方式,通过下拉列表选择计算求和的项数(如图 8-8 所示)。





图 8-8 获取 SELECT 提交的值,计算数列和

```
select.jsp:
<HTML>
<%@ page contentType = "text/html;charset = GB2312" %>
<BODY bgcolor = cyan><FONT size = 1 >
<P>选择计算和的方式
   <FORM action = "sum.jsp" method = post name = form>
        <SELECT name = "sum" size = 2 >
            <OPTION Selected value = "1">计算 1 到 n 的连续和
            <OPTION value = "2">计算 1 到 n 的平方和
            <OPTION value = "3">计算 1 到 n 的立方和
        </SELECT>
<P>选择 n 的值: <BR>
        <SELECT name = "n" >
```

```
            <OPTION value="10">n=10
            <OPTION value="20">n=20
            <OPTION value="30">n=30
            <OPTION value="40">n=40
            <OPTION value="50">n=50
            <OPTION value="100">n=100
         </SELECT>
          <BR><BR>
      <INPUT type="submit" value="提交你的选择" name="submit">
    </FORM></FONT></BODY></HTML>
sum.jsp:
<HTML>
<%@ page contentType="text/html;charset=GB2312" %>
<BODY bgcolor=cyan><Font size=1>
 <% long sum=0;
    String s1=request.getParameter("sum");
    String s2=request.getParameter("n");
     if(s1==null)
       {s1="";}
     if(s2==null)
       {s2="0";}
    if(s1.equals("1"))
       {int n=Integer.parseInt(s2);
         for(int i=1;i<=n;i++)
             {sum=sum+i; }}
      else if(s1.equals("2"))
       {int n=Integer.parseInt(s2);
         for(int i=1;i<=n;i++)
             {sum=sum+i*i; }}
      else if(s1.equals("3"))
       {int n=Integer.parseInt(s2);
         for(int i=1;i<=n;i++)
             {sum=sum+i*i*i; } } %>
<P>你的求和结果是<%=sum%></FONT></BODY></HTML>
```

3. <TEXTAREA>格式

<TEXTAREA>标记在表单中指定一个能输入多行文本的文本区域。

```
<TEXTAREA  name="ilovethisgame" Rows="4" Cols="20">
</TEXTAREA>
```

4. 表格

表格由<TABLE>、</TABLE>标记定义，一般格式如下：

```
<TABLE>
    <TR width="该行的宽度">
        <TH width="单元格的宽度">单元格中的数据</TH>
         ...
```

```
    <TD width= "单元格的宽度" >单元格中的数据</TD> ...
</TR>...</TABLE>
```

其中,<TR>...</TR>定义表格的一个行。<TH>和<TD>标记定义这一行中的表格单元,二者的区别是:<TH>定义的单元着重显示,<TD>称做普通单元,不着重显示;一行中的着重单元和普通单元可以交替出现,也可以全是着重单元或普通单元。

在<TABLE>中增加选项 Border 可指明该表格是否带有边框。

在下面的例 8-8 中,用一个 3 行的表格显示数据。第一行有 3 个着重显示的单元;第二行有 3 个单元,其中第一个单元着重显示;第三行有 3 个普通的单元,如图 8-9 所示。



图 8-9 用表格显示数据

【例 8-8】 用表格显示数据(如图 8-9 所示)。

```
table.jsp:
<HTML>
<%@ page contentType="text/html;charset=GB2312" %>
<BODY><TABLE align="Center" Border>
    <TR width=400>
        <TH  Align="Center">中间</TH>
        <TH  Align="Right">右</TH>
        <TH  Align="LEFT">左</TH>
          <TD></TD>
        <TD></TD>
    </TR>  <TR>
        <TH  Valign="Top">数据靠向上沿</TH>
        <TD  Valign="Bottom">数据靠向下沿</TD>
        <TD  Valign="Bottom" Align="Center" >数据居中靠向下沿</TD></TR>
    <TR><TD  Valign="Top"> 你好</TD>
        <TD  Valign="Bottom"> hello</TD>
        <TD  Valign="Bottom" Aligin="Center" >112334 </TD>
    </TR></TABLE></BODY></HTML>
```

在下面的例 8-9 中,把一个表单显示在表格的一个单元格中。

【例 8-9】 把一个表单显示在表格的一个单元格中(如图 8-10 所示)。

```
tableform.jsp:
<HTML>
<%@ page contentType="text/html;charset=GB2312" %>
```

```
<BODY bgcolor = cyan>
  <TABLE align = "left" Border>
     <TR><TH  width = 250><FONT size = 1>李白是哪个朝代的人? </TH>
            <TD width = 220>
              <FORM action = "answer.jsp" method = post name = form>
                <INPUT type = "radio" name = "R" value = "a"  ><FONT size = 1>宋朝
                <INPUT type = "radio" name = "R" value = "b"  ><FONT size = 1>唐朝
                <INPUT type = "submit" name = "g" value = "送出">
              </FORM></TD></TR><TR>
            <TH><FONT size = 1>输入数据: </TH>
        <FORM action = "answer.jsp" method = post name = form>
          <TD>  <INPUT type = "text" name = "R" value = "a" size = 20></TD>
          <TD>  <INPUT type = "submit" name = "f" value = "送出"></TD></FORM>
     </TR>  </TABLE></BODY></HTML>
```



图 8-10 在表格中显示表单数据

8.2 response 对象

当客户访问一个服务器的页面时,会提交一个 HTTP 请求,服务器收到请求时,返回 HTTP 响应。响应和请求类似,也有某种结构,每个响应都由状态行开始,可以包含几个头及可能的信息体(网页的结果输出部分)。

8.1 节学习了用 request 对象获取客户请求提交的信息,与 request 对象相对应的对象是 response 对象。可以用 response 对象对客户的请求作出动态响应,向客户端发送数据。

8.2.1 动态响应 contentType 属性

当一个客户请求访问一个 JSP 页面时,如果该页面用 page 指令设置页面的 contentType 属性的值是 text/html,那么 JSP 引擎将按照这种属性值作出响应,将页面的静态部分返回给客户。由于 page 指令只能为 contentType 指定一个值,来决定响应的 MIME 类型,如果想动态地改变这个属性的值来响应客户,就要使用 response 对象的 setContentType(String s) 方法来改变 contentType 的属性值:

```
public void setContentType(String  s);
```

该方法动态设置响应的 MIME 类型,参数 s 可取 text/html、text/plain、application/x-

msexcel、application/msword 等。

8.2.2 response 的 HTTP 文件头

从 8.1.3 小节的内容知道，一个典型的请求通常包含很多的头，有些头是标准的，有些和特定的浏览器有关。同样，响应也包括一些头。response 对象可以使用方法

```
addHeader(String  head,String value);
```

或方法

```
setHeader(String head ,String value)
```

动态添加新的响应头和头的值，将这些头发送给客户的浏览器。如果添加的头已经存在，则先前的头被覆盖。

8.2.3 response 重定向

在某些情况下，当响应客户时，要将客户重新引导至另一个页面。例如，如果客户输入的表单信息不完整，就会再被引导到该表单的输入页面。

可以使用 response 的 sendRedirect(URL url)方法实现客户的重定向。

8.2.4 response 的状态行

当服务器对客户请求进行响应时，它发送的首行称做状态行。

状态行包括 3 位数字的状态代码和对状态代码的描述(称做原因短语)。下面列出了对 5 类状态的代码的大概描述。

1yy(1 开头的 3 位数)：主要是实验性质的。例如，101 表示服务器正在升级协议。

2yy：用来表明请求成功。例如，状态代码 200 可以表明已成功取得了请求的页面。

3yy：用来表明在请求满足之前应采取进一步的行动。例如，305 表示请求必须通过代理来访问。

4yy：当浏览器做出无法满足的请求时，返回该状态代码。例如，404 表示请求的页面不存在。

5yy：用来表示服务器出现问题。例如，500 说明服务器内部发生错误。

一般无须修改状态行，在出现问题时，服务器会自动响应，发送相应的状态代码。也可以使用 response 对象的 setStatus(int n)方法来增加状态行的内容。在下面的例 8-10 中，使用 setStatus(int n)方法设置响应的状态行。

【例 8-10】 使用 setStatus(int n)方法设置响应的状态行(如图 8-11 所示)。



图 8-11 设置响应状态行

```
Example8_14.jsp:
<%@ page contentType="text/html;charset=GB2312" %>
```

```
<HTML><BODY bgcolor=cyan><Font size=1>
<P>点击下面的超链接:<BR>
  <A HREF="bird1.jsp">bird1:欢迎你吗?<BR>
  <A HREF="bird2.jsp">bird2:欢迎你吗?<BR>
  <A HREF="bird3.jsp">bird3:欢迎你吗?</FONT></BODY></HTML>
bird1.jsp:
<HTML><BODY>
<% response.setStatus(408);
   out.print("不显示了"); %></BODY></HTML>
bird2.jsp:
<%@ page contentType="text/html;charset=GB2312" %><HTML><BODY>
<% response.setStatus(200);
   out.println("ok");   %></BODY></HTML>
bird3.jsp:
<%@ page contentType="text/html;charset=GB2312" %><HTML><BODY>
<%  response.setStatus(500); %></BODY></HTML>
```

表 8-1 是状态代码表。

表 8-1 状态代码表

状态代码	代码说明	状态代码	代码说明
101	服务器正在升级协议	404	请求的资源不可用
100	客户可以继续	405	请求所用的方法是不允许的
201	请求成功且在服务器上创建了新的资源	406	请求的资源只能用请求不能接受的内容特性来响应
202	请求已被接受但还没有处理完毕	407	客户必须得到认证
200	请求成功	408	请求超时
203	客户端给出的元信息不是发自服务器的	409	发生冲突,请求不能完成
204	请求成功,但没有新信息	410	请求的资源已经不可用
205	客户必须重置文档视图	411	请求需要一个定义的内容长度才能处理
206	服务器执行了部分 get 请求	413	请求太大,被拒绝
300	请求的资源有多种表示法	414	请求的 URL 太大
301	资源已经被永久移动到新位置	415	请求的格式被拒绝
302	资源已经被临时移动到新位置	500	服务器发生内部错误,不能服务
303	应答可以在另外一个 URL 中找到	501	不支持请求的部分功能
304	Get 方式请求不可用	502	从代理和网关接受了不合法的字符
305	请求必须通过代理来访问	503	HTTP 服务暂时不可用
400	请求有语法错误	504	服务器在等待代理服务器应答时发生超时
401	请求需要 HTTP 认证	505	不支持请求的 HTTP 版本
403	取得了请求但拒绝服务		

8.3 session 对象

HTTP 协议是一种无状态协议。一个客户向服务器发出请求(request)然后服务器返回响应(response),连接就被关闭了。在服务器端不保留连接的有关信息,因此当下一次连接时,服务器已没有以前的连接信息了,无法判断这一次连接和以前的连接是否属于同一客户。因此,必须使用会话记录有关连接的信息。

从一个客户打开浏览器连接到服务器，到客户关闭浏览器离开这个服务器，称做一个会话。当一个客户访问一个服务器时，可能会在这个服务器的几个页面反复连接、反复刷新一个页面或不断地向一个页面提交信息等，服务器应当通过某种办法知道这是同一个客户，这就需要 session(会话)对象。

8.3.1 session 对象的 ID

当一个客户首次访问服务器上的一个 JSP 页面时，JSP 引擎产生一个 session 对象，这个 session 对象调用相应的方法可以存储客户在访问各个页面期间提交的各种信息，比如姓名、号码等信息。这个 session 对象被分配了一个 String 类型的 ID 号，JSP 引擎同时将这个 ID 号发送到客户端，存放在客户的 Cookie 中。这样，session 对象和客户之间就建立起一一对应的关系，即每个客户都对应着一个 session 对象(该客户的会话)，这些 session 对象互不相同，具有不同的 ID 号码。我们已经知道，JSP 引擎为每个客户启动一个线程，也就是说，JSP 为每个线程分配不同的 session 对象。当客户再访问连接该服务器的其他页面时，或从该服务器连接到其他服务器再回到该服务器时，JSP 引擎不再分配给客户新的 session 对象，而是使用完全相同的一个，直到客户关闭浏览器后，服务器端该客户的 session 对象被取消，和客户的会话对应关系消失。当客户重新打开浏览器再连接到该服务器时，服务器为该客户再创建一个新的 session 对象。

在下面的例 8-11 中，客户在服务器的 3 个页面之间进行连接，只要不关闭浏览器，3 个页面的 session 对象是完全相同的。客户首先访问 session.jsp 页面，从这个页面再连接到 tom.jsp 页面，然后从 tom.jsp 页面再连接到 jerry.jsp 页面。

【例 8-11】 客户在服务器的 3 个页面之间进行连接(如图 8-12 所示)。

```
session.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML><BODY><P>
   <% String s = session.getId(); %>
<P> 你的 session 对象的 ID 是:
   <BR>    <% = s %>
<P>输入你的姓名连接到 tom.jsp
   <FORM action = "tom.jsp" method = post name = form>
        <INPUT type = "text" name = "boy">
        <INPUT type = "submit" value = "送出" name = submit>
  </FORM>  </BODY></HTML>
tom.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML><BODY>
<P>我是 Tom 页面
   <% String s = session.getId();   %>
<P> 你在 Tom 页面中的 session 对象的 ID 是:   <% = s %>
<P> 点击超链接,连接到 Jerry 的页面。
<A HREF = "jerry.jsp">
<BR>  欢迎到 Jerry 屋来!</A>
</BODY></HTML>
jerry.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML><BODY>
```

```
<P>我是 Jerry 页面
  <% String s = session.getId();  %>
<P>你在 Jerry 页面中的 session 对象的 ID 是:<% = s %>
<P>点击超链接,连接到 session 的页面。
<A HREF = "session.jsp">
<BR> 欢迎到 session 屋来!</A></BODY></HTML>
```

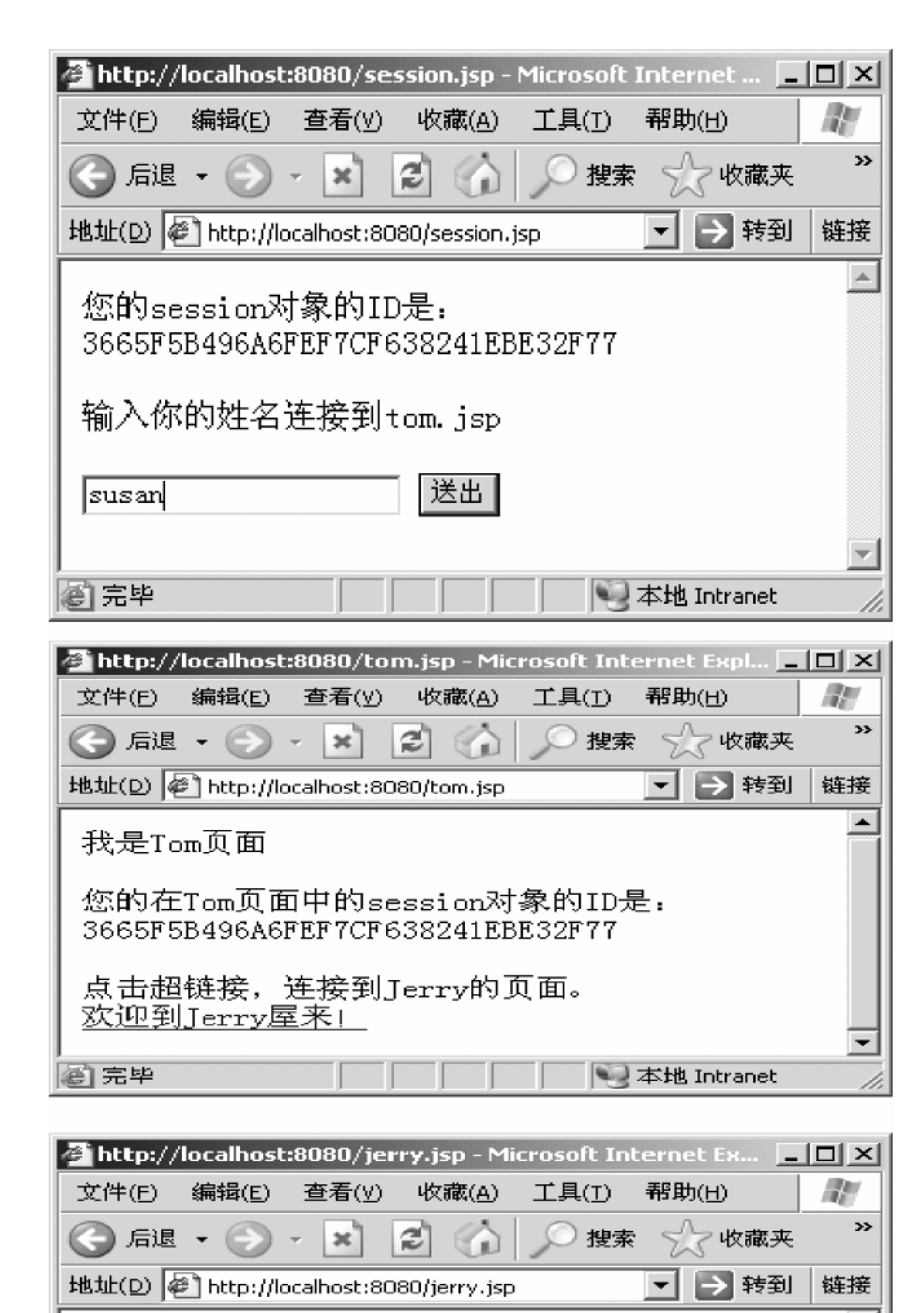


图 8-12 session 对象的 ID

8.3.2 session 对象与 URL 重写

session 对象能否和客户建立起一一对应关系依赖于客户的浏览器是否支持 Cookie。如果客户的浏览器不支持 Cookie，那么客户在不同网页之间的 session 对象可能是互不相同的，因为服务器无法将 ID 存放到客户端，就不能建立 session 对象和客户的一一对应关系。将浏览器的 Cookie 设置为禁止后(选择浏览器菜单→“工具”→“Internet 选项”→“安全”→Internet 和“本地 Intranet”→“自定义级别”→Cookie，将全部选项设置成禁止)，也就是说，“同一客户”对应了多个 session 对象，这样服务器就无法知道在这些页面上访问的实际上是同一个客户。

如果客户的浏览器不支持 Cookie，可以通过 URL 重写来实现 session 对象的唯一性。所谓 URL 重写，就是当客户从一个页面重新连接到另一个页面时，通过向这个新的 URL 添加参数，把 session 对象的 ID 传带过去，这样就可以保障客户在该网站各个页面中的 session 对象是完全相同的。可以使用 response 对象调用 encodeURL()或 encodeRedirect URL()方法实现 URL 重写，比如，如果从 Tom 页面连接到 Jerry 页面，首先实现 URL 重写：

```
String str = response.encodeRedirectURL("jerry.jsp");
```

然后将连接目标写成<%=str%>。

8.3.3 session 对象的常用方法

1. public void setAttribute(String key,Object obj)

session 对象类似于散列表，session 对象可以调用该方法将参数 Object 指定的对象 obj 添加到 session 对象中，并为添加的对象指定一个索引关键字，如果添加的两个对象的关键字相同，则先前添加的对象被清除。

2. public Object getAttibue(String key)

获取 session 对象含有的关键字是 key 的对象。由于任何对象都可以添加到 session 对象中，因此用该方法取回对象时，应强制转化为原来的类型。

3. public Enumeration getAttributeName()

session 对象调用该方法产生一个枚举对象，该枚举对象使用 nextElemets()遍历 session 对象所含有的全部对象。

4. public long getCreationTime()

session 对象调用该方法可以获取该对象创建的时间，单位是毫秒(从 1970 年 7 月 1 日午夜起至该对象创建时刻所走过的毫秒数)。

5. public long getLastAccessedTime()

获取当前 session 对象最后一次被操作的时间，单位是毫秒。

6. public int getMaxInactiveIterval()

获取 session 对象的生存时间。

7. public void setMaxInactiveIterval(int　n)

设置 session 对象的生存时间(单位是秒)

8. public void removeAttribue(String key)

从当前 session 对象中删除关键字是 key 的对象。

9. public String getId()

获取 session 对象的编号。

10. invalidate

使得 session 无效。

8.3.4　计数器

下面的【例 8-12】中,用 session 对象禁止客户通过刷新页面增加计数。当客户刷新页面时,可以使用 session 的 public boolean isNew()方法判断是不是一个新的客户,因为客户刷新页面不会改变服务器分配给该客户的 session 对象。

【例 8-12】　计数器实验。

```
Examle8_19.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<%@ page import = "java.io.*" %>
<HTML>
<BODY>
    <%! int number = 0;
     synchronized void countPeople()
        {if(number == 0)
            { try{File f = new File("D:/tomcat","countPeople.txt");
                 FileInputStream in = new FileInputStream(f);
                 DataInputStream dataIn = new DataInputStream(in);
                 number = dataIn.readInt();
                 number ++ ;
                 in.close();dataIn.close(); }
                 catch(FileNotFoundException e)
                  { number ++ ;
                   try {File f = new File("D:/tomcat","countPeople.txt");
                        FileOutputStream out = new FileOutputStream(f);
                        DataOutputStream dataOut = new DataOutputStream(out);
                        dataOut.writeInt(number);
                        out.close();dataOut.close(); }
                    catch(IOException ee){} }
               catch(IOException ee)
```

```
                    {  } }
              else
                {number ++ ;
                try{File f = new File("D:/tomcat","countPeople.txt");
                    FileOutputStream out = new FileOutputStream(f);
                    DataOutputStream dataOut = new DataOutputStream(out);
                    dataOut.writeInt(number);
                    out.close();dataOut.close(); }
                catch(FileNotFoundException e){}
                catch(IOException e){}}} %><%
         if(session.isNew())
          {countPeople();
            String str = String.valueOf(number);
            session.setAttribute("count",str); } %>
<P>您是第<% = (String)session.getAttribute("count") %>个访问本站的人.
<BODY><HTML>
```

8.4 application 对象

我们已经知道，当客户第一次访问服务器上的一个 JSP 页面时，JSP 引擎创建一个和该客户相对应的 session 对象，当客户在所访问的网站的各个页面之间浏览时，这个 session 对象都是同一个，直到客户关闭浏览器，这个 session 对象才被取消；而且不同客户的 session 对象是互不相同的。与 session 对象不同的是 application 对象。服务器启动后，就产生了这个 application 对象。当客户访问服务器上的一个 JSP 页面时，JSP 引擎为该客户分配这个 application 对象，当客户在所访问的网站的各个页面之间浏览时，这个 application 对象都是同一个，直到服务器关闭，这个 application 对象才被取消。与 session 对象不同的是，所有客户的 application 对象是相同的一个，即所有的客户共享这个内置的 application 对象。JSP 引擎为每个客户启动一个线程，也就是说，这些线程共享这个 application 对象。

8.4.1 application 对象的常用方法

1. public void setAttribute(String key,Object obj)

application 对象可以调用该方法将参数 Object 指定的对象 obj 添加到 application 对象中，并为添加的对象指定一个索引关键字，如果添加的两个对象的关键字相同，则先前添加对象被清除。

2. public Object getAttibue(String key)

该方法获取 application 对象含有的关键字是 key 的对象。由于任何对象都可以添加到 application 对象中，因此用该方法取回对象时，应强制转化为原来的类型。

3. public Enumeration getAttributeNames()

application 对象调用该方法产生一个枚举对象，该枚举对象使用 nextElemets()遍历

application 对象所含有的全部对象。

4. public void removeAttribute(String key)

该方法从当前 application 对象中删除关键字是 key 的对象。

5. public String getServletInfo()

该方法获取 Servlet 编译器的当前版本的信息。

由于 application 对象对所有的客户都是相同的，任何客户对该对象中存储的数据的改变都会影响到其他客户，因此在某些情况下，对该对象的操作需要实现同步处理。

application 对象实现一个计数器，将计数存放在 application 对象中，每个客户对该对象中"计数"的改变都会影响到其他客户。

注：*有些服务器不直接支持使用 application 对象，必须用 ServletContext 类声明这个对象，再使用 getServletContext()方法对这个 application 对象进行初始化。*

【例 8-13】 利用 application 进行计数。

```
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML><BODY>
    <%! synchronized void countPeople()
        { ServletContext  application = getServletContext();
          Integer number = (Integer)application.getAttribute("Count");
          if(number == null)
              { number = new Integer(1);
                application.setAttribute("Count",number); }
        else
              { number = new Integer(number.intValue() + 1);
                application.setAttribute("Count",number); } } %>
    <% if(session.isNew())
       {  countPeople();
          Integer myNumber = (Integer)application.getAttribute("Count");
          session.setAttribute("MyCount",myNumber);  } %>
<P><P>您是第
  <% int a = ((Integer)session.getAttribute("MyCount")).intValue();
   %><% = a %>个访问本站的客户。</BODY></HTML>
```

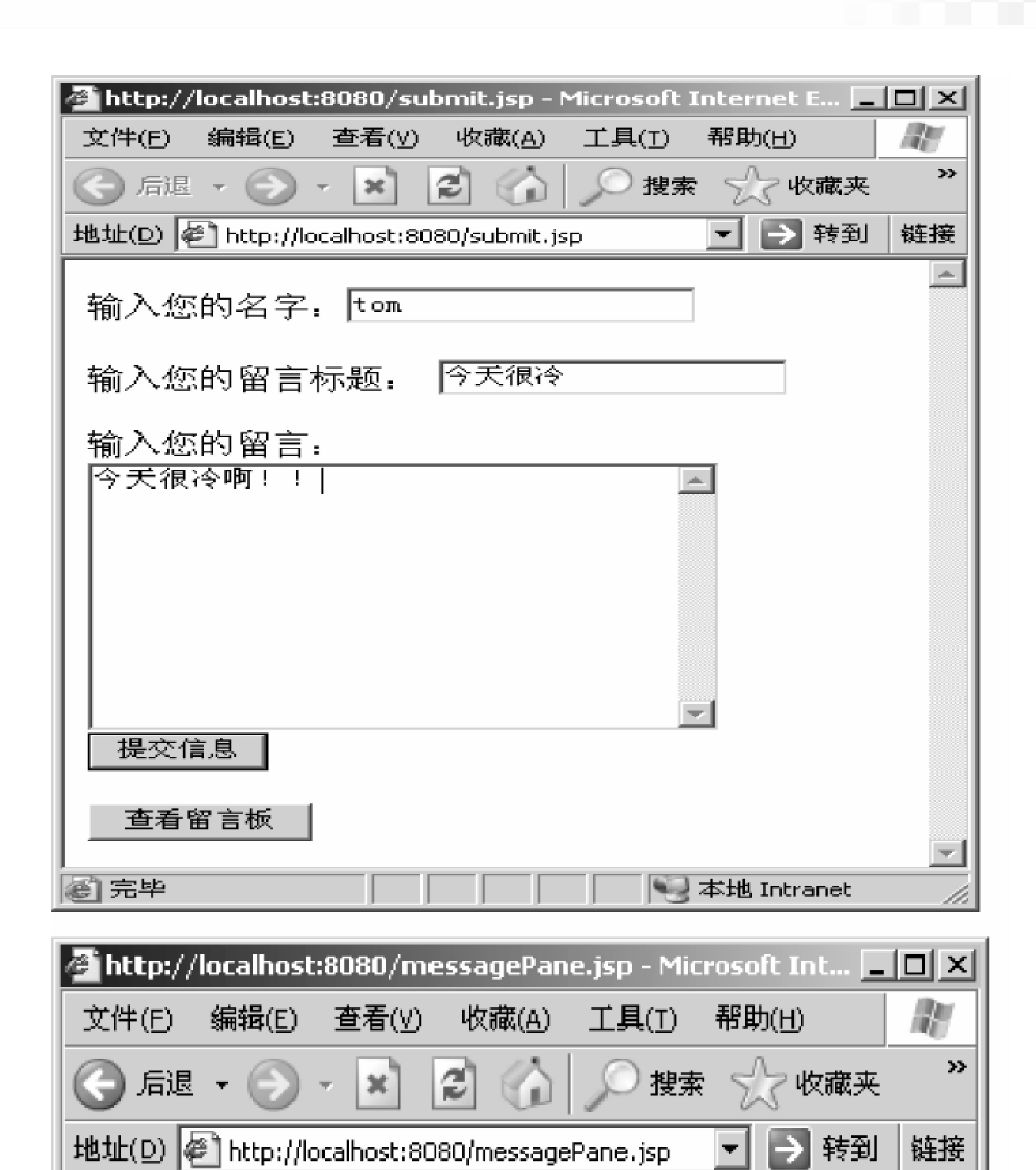
8.4.2　用 application 制作留言板

在例 8-14 中，客户通过 submit.jsp 向 messagePane.jsp 页面提交姓名、留言标题和留言内容，messagePane.jsp 页面获取这些内容后，用同步方法将这些内容添加到一个向量中，然后将这个向量再添加到 application 对象中。当用户点击查看留言板时，showMessage.jsp 负责显示所有客户的留言内容，即从 application 对象中取出向量，然后遍历向量中存储的信息。

【例 8-14】 用 application 制作留言板(如图 8-13 所示)。

```
submit.jsp:
<%@ page contentType = "text/html;charset = GB2312" %>
<HTML><BODY><FORM action = "messagePane.jsp" method = "post" name = "form">
```

```
      <P>输入您的名字:<INPUT  type="text" name="peopleName">  <BR>
       <P>输入您的留言标题:
      <INPUT  type="text"  name="Title">  <BR>
       <P>输入您的留言:<BR>
      <TEXTAREA name="messages" ROWs="10" COLS=36 WRAP="physical">
      </TEXTAREA>
      <BR><INPUT type="submit" value="提交信息" name="submit">
</FORM>
<FORM action="showMessage.jsp" method="post" name="form1">
<INPUT type="submit" value="查看留言板" name="look"></FORM></BODY></HTML>
messagePane.jsp:
<%@ page contentType="text/html;charset=GB2312" %>
<%@ page import="java.util.*" %>
<HTML><BODY>
    <%! Vector v=new Vector();
         int i=0; ServletContext  application;
         synchronized void sendMessage(String s)
           { application=getServletContext();;
             i++;
           v.add("No."+i+","+s);
           application.setAttribute("Mess",v); } %>
    <% String name=request.getParameter("peopleName");
        String title=request.getParameter("Title");
        String messages=request.getParameter("messages");
           if(name==null)
             {name="guest"+(int)(Math.random()*10000);}
           if(title==null)
             {title="无标题"; }
           if(messages==null)
             {messages="无信息"; }
String s="Name:"+name+"#"+"Title:"+title+"#"+"Content:"+"<BR>"+messages;
          sendMessage(s);
          out.print("您的信息已经提交!");   %>
  <A HREF="submit.jsp">返回 </BODY></HTML>
showMessage.jsp:
<%@ page contentType="text/html;charset=GB2312" %>
<%@ page import="java.util.*" %>
<HTML><BODY>
    <% Vector v=(Vector)application.getAttribute("Mess");
          for(int i=0;i<v.size();i++)
            { String message=(String)v.elementAt(i);
              StringTokenizer fenxi=new StringTokenizer(message,"#");
                while(fenxi.hasMoreTokens())
                     { String str=fenxi.nextToken();
                         byte a[]=str.getBytes("ISO-8859-1");
                         str=new String(a);
                       out.print("<BR>"+str); } } %></BODY></HTML>
```


http://localhost:8080/submit.jsp - Microsoft Internet E...
文件(F)　编辑(E)　查看(V)　收藏(A)　工具(T)　帮助(H)
地址(D) http://localhost:8080/submit.jsp
输入您的名字：tom
输入您的留言标题：今天很冷
输入您的留言：
今天很冷啊！！
提交信息
查看留言板
完毕　本地 Intranet
http://localhost:8080/messagePane.jsp - Microsoft Int...
地址(D) http://localhost:8080/messagePane.jsp
您的信息已经提交！　返回
完毕　本地 Intranet


http://localhost:8080/showMessage.jsp - Microsoft Int...
地址(D) http://localhost:8080/showMessage.jsp
No.1,Name:tom
Title:今天很冷
Content:
今天很冷啊！！
完毕　本地 Intranet

图 8-13　留言板

8.5 out 对象

out 对象是一个输出流，用来向客户端输出数据。在前面的许多例子里曾多次使用 out 对象进行数据的输出。out 对象可调用如下的方法用于各种数据的输出。

- out.print(Boolean)，out.println(boolean)：输出一个布尔值。
- out.print(char)，out.println(char)：输出一个字符。
- out.print(double)，out.println(double)：输出一个双精度的浮点数。
- out.print(fload)，out.println(float)：输出一个单精度的浮点数。
- out.print(long)，out.println(long)：输出一个长整型数据。
- out.print(String)，out.println(String)：输出一个字符串对象的内容。
- out.newLine()：输出一个换行符。
- out.flush()：输出缓冲区里的内容。
- out.close()：关闭流。

下面的例 8-15 使用 out 对象向客户输出包括表格等内容的信息。

【例 8-15】 使用 out 对象向客户输出包括表格等内容的信息(如图 8-14 所示)。

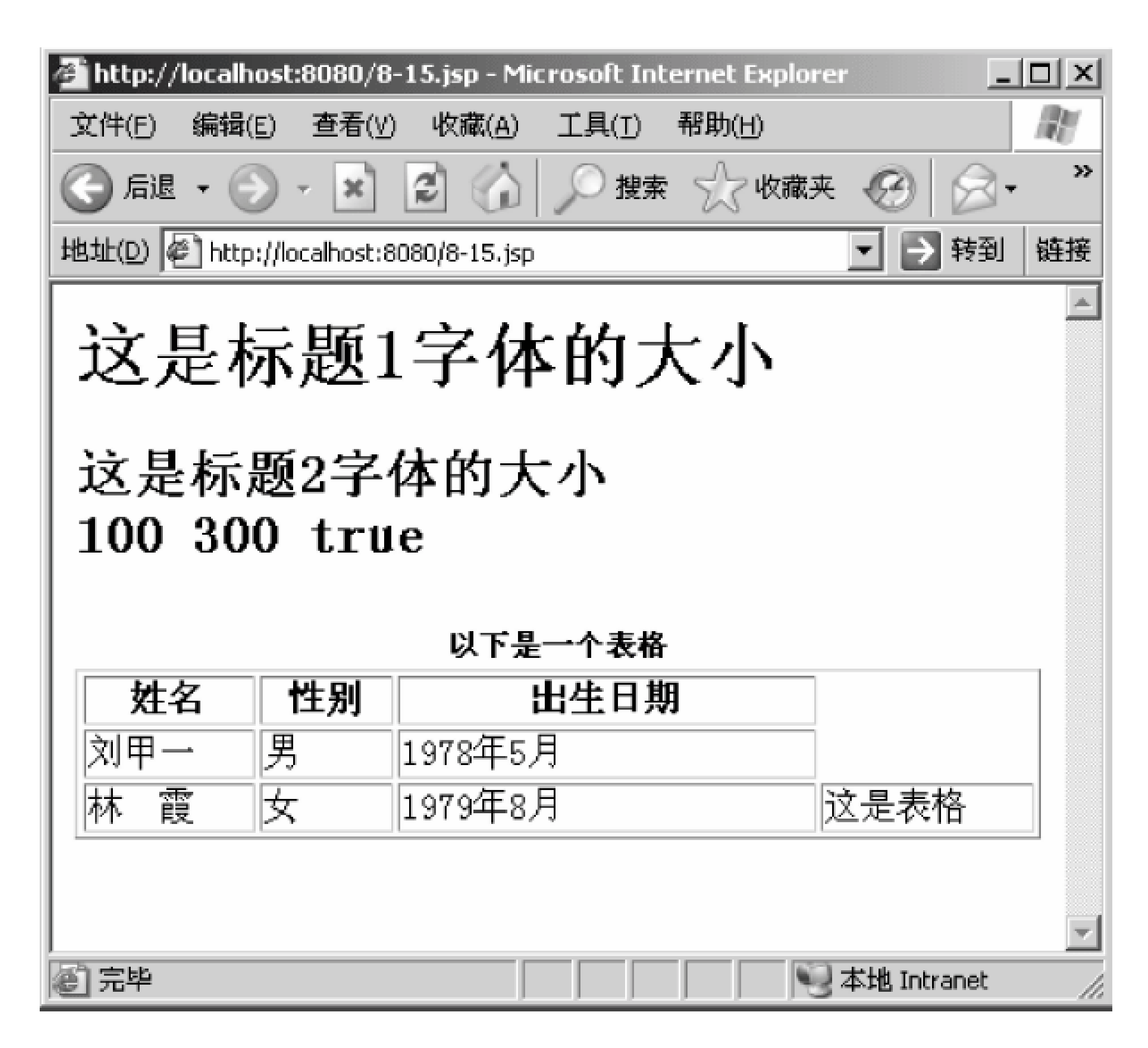


图 8-14 用 out 对象输出各种信息

```
Example8_22.jsp:
<%@ page contentType="text/html;charset=GB2312" %>
<%@ page import="java.util.*" %>
<HTML><BODY>
  <% int a=100;long b=300;boolean c=true;
     out.println("<H1>这是标题 1 字体的大小</HT1>");
     out.println("<H2>这是标题 2 字体的大小</HT2>");
     out.print("<BR>");
```

```
    out.println(a);  out.println(b);  out.println(c); %>
<CENTER><P><FONT size=2>以下是一个表格</Font>
<% out.print("<FONT face=隶书 size=2>");
     out.println("<TABLE Border>");
       out.println("<TR>");
           out.println("<TH width=80>"+"姓名"+"</TH>");
           out.println("<TH width=60>"+"性别"+"</TH>");
           out.println("<TH width=200>"+"出生日期"+"</TH>");
       out.println("</TR>");
       out.println("<TR>");
           out.println("<TD>"+"刘甲一"+"</TD>");
           out.println("<TD>"+"男"+"</TD>");
           out.println("<TD>"+"1978 年 5 月"+"</TD>");
       out.println("</TR>");
       out.println("<TR>");
           out.println("<TD>"+"林 霞"+"</TD>");
           out.println("<TD>"+"女"+"</TD>");
           out.println("<TD>"+"1979 年 8 月"+"</TD>");
            out.println("<TD width=100>"+"这是表格"+"</TD>");
       out.println("</TR>");
    out.println("</TABLE>");
   out.print("</FONT>")   %>  </CENTER></BODY></HTML>
```

下面的例 8-16 根据当前时间向客户发送提示信息，比如，如果是 26 日，就发送病毒易发作的信息等。

【例 8-16】 根据当前时间向客户发送提示信息(如图 8-15 所示)。



图 8-15　发送提示信息

```
time.jsp:
<%@ page contentType="text/html;charset=GB2312" %>
<%@ page import="java.util.*" %>
<%! public  String  getDayWeek(int n)
{ String week[]={"星期日","星期一","星期二","星期三","星期四","星期五","星期六"};
      return week[n]; } %>
<HTML><BODY bgcolor=cyan><FONT size=1>
<% Calendar calendar=Calendar.getInstance();      //创建一个日历对象
           calendar.setTime(new Date());          //用当前时间初始化日历时间
   String 年=String.valueOf(calendar.get(Calendar.YEAR)),
          月=String.valueOf(calendar.get(Calendar.MONTH)+1),
          日=String.valueOf(calendar.get(Calendar.DAY_OF_MONTH)),
```

```
        星期 = getDayWeek(calendar.get(Calendar.DAY_OF_WEEK) - 1);
  int hour = calendar.get(Calendar.HOUR_OF_DAY),
      minute = calendar.get(Calendar.MINUTE),
      second = calendar.get(Calendar.SECOND); %>
<P>现在的时间是<BR>
<% = 年 %>年
<% = 月 %>月
<% = 日 %>日
<% = 星期 %><BR>
<% = hour %>时
<% = minute %>分
<% = second %>秒
<%  if(日.equals("26"))
    {out.print("<BR><H2>今天是病毒容易发作的日子!</H2>"); }
    if(hour >= 22)
    { out.print("<BR><H2>现在时间很晚了注意休息</H2>");} %>
</FONT></BODY></HTML>
```

8.6 本章小结

本章介绍了 JSP 的 5 种内置对象，给出了每一个对象所具有的方法及相应的使用说明。熟练使用这些内置对象是开发 JSP 应用程序的基本要求，尤其是对于 request、response、application、session 和 out 对象要熟练掌握。

out 对象是 javax.servlet.jsp.JspWriter 类的一个子类的对象，它的作用是把信息传送到客户机的浏览器中。作用域：当前页面。常用的方法有：① print()：输出内容；②println()：输出内容并换行；③newLine()：输出一个换行符；④flush()：输出缓冲的内容；⑤write()：与 print()相同；⑥close()：关闭输出流。

request 对象是 javax.servlet.HttpServletRequest 类的子类的对象，当客户端请求一个 JSP 页面时，JSP 容器会将客户端的请求信息封装在这个 request 对象中(请求信息的内容包括请求的头信息(Header)、系统信息(比如编码方式)、请求的方式(比如 GET 或 POST)、请求的参数名称和参数值等信息)。作用域：待定。常用的方法：①getParameter(String parameterName)：获取浏览器请求的参数值；②getParameterNames()：获取浏览器传递过来的所有参数名；③setCharacterEncoding(String encodingName)：当利用 getParameter(String parameterName)获得表单(针对 post 方法)中元素的时候，在执行操作之前，设置 request 的编码格式，例如“request.setCharacterEncoding("GBK")”；④setAttribute(String name,Object value)：设置属性和属性值；⑤getAttribute(String name)：获得属性值。

response 对象是一个 javax.servlet.http.HttpServletResponse 类的子类的对象。它的作用是对客户端请求的响应。作用域：当前页面。常用的方法有：①sendRedirect(String url)：重定向，不支持数据共享；②setContentType(String contentType)：指定 HTTP 响应和浏览器显示的编码类型；③setHeader(String name,String value)：自动跳转。

session 对象是 java.servlet.http.HttpSession 类的子类的对象，它表示当前的用户会话信息。在 session 中保存当前用户连接的所有页面中的相关信息。作用域：会话期间。

常用的方法有：①setAttribute(String name,Object value)：设定一个属性和属性值；②getAttribute(String name)：获取指定属性name的属性值；③invalidate()：销毁session。

8.7　思考与练习

1. JSP内置对象的作用域由小到大的排列顺序是什么？
2. JSP内置对象包括哪些内容？
3. 内置对象的page指令用法是什么？
4. application对象的作用域是什么？
5. 内置对象page和pageContext有什么不同？
6. out.flush()与out.clear()有什么区别？
7. request. GetAttribute(String name)的作用是什么？
8. response. AddHeader(String name，String value)方法的作用是什么？
9. JSP内置对象与Servlet API的关系是什么？

第三篇

数据库开发

第9章 Web数据库技术

数据库技术产生于20世纪60年代末70年代初，它的出现使计算机应用进入了一个新的时期——社会的每一个领域都与计算机应用发生了联系。数据库是计算机的最重要的技术之一，是计算机软件的一个独立分支，数据库是建立管理信息系统的核心技术，当数据库与网络通信技术、多媒体技术结合在一起时，计算机应用将无所不在、无所不能。

本章主要介绍Access数据库和SQL Server数据库，并从这两个数据库软件中学习数据库的基本操作方法，创建数据库、表、查询等操作方法，学习如何将数据库技术应用到Web开发中。

学习目标及要求

(1) 熟悉创建Access数据库和表的方法。

(2) 掌握表之间的关系。

(3) 能够使用Access查询数据。

(4) 学会安装SQL Server服务器，能够启动、关闭服务。

(5) 熟练掌握SQL语句，包括创建、查询、插入、修改、删除语句。

(6) 学会备份数据库的方法。

(7) 学会各种导入和导出Access和SQL Server数据的方法。

9.1 Access数据库

Access 2003是一个功能强大、方便灵活的关系型数据库管理系统。Access 2003作为一个小型数据库管理系统，为用户提供了一个数据库管理工具集和数据库应用系统开发平台。

Access启动方法：双击桌面Access快捷方式图标。退出方法：单击主窗口中的关闭按钮。本节主要侧重于从Web数据库开发角度讲解Access数据库功能。

9.1.1 创建数据库

进入Access，打开一个示例数据库，可以看到如图9-1所示的界面，在这个界面的“对象”栏中，包含Access的7个对象，分别为表、查询、窗体、报表、页、宏、模块。

创建数据库方法：双击Access图标，选择“文件”→“新建”→“空数据库”。默认文件名为db1.mdb。Access数据库文件的后缀名为.mdb。



图 9-1 学生管理系统数据库

9.1.2 设计表

1. Access 数据表对象组成

Access 数据表对象由表结构和表数据两个部分构成。数据表对象的结构是指数据表的框架，也称为数据表对象的属性，主要包括以下内容：

(1) 字段名称。数据表中的一列称为一个字段，而每一个字段均具有唯一的名字，称为字段名称。

(2) 数据类型。数据表中的同一列数据必须具有共同数据特征，称为字段的数据类型。

(3) 字段大小。数据表中的一列所能容纳的字符或数字的个数称为字段大小。

(4) 字段的其他属性，包括索引、格式等。

打开要创建数据表的数据库，单击“数据库”窗口对象栏中的“表”按钮，然后单击“数据库”窗口工具栏中的“新建”按钮，打开“新建表”对话框，选择“设计视图”，然后单击“确定”按钮，打开表设计视图，如图 9-2 所示。



图 9-2 表结构

打开要创建数据表的数据库，单击“数据库”窗口对象栏中的“表”按钮，然后单击“数据库”窗口工具栏中的“新建”按钮，打开“新建表”对话框，选择“数据表视图”，并单击“确定”按

钮,打开“数据表”视图窗口,在该窗口下可以通过输入数据直接创建表,如图 9-3 所示。

学生学籍表 : 表

学号	姓名	班级	专业	联系电话	家庭住址
4z081101	贾琛	1班	计算机科学与技术	18966600011	陕西西安
4z081102	卢伟	1班	计算机科学与技术	18966500012	陕西咸阳
4z081103	葛文隽	1班	计算机科学与技术	18966200013	陕西临潼
4z081104	林青	1班	计算机科学与技术	18966300014	陕西渭南
4z081105	黄玉蕾	1班	计算机科学与技术	18966500015	河南郑州
4z081106	王京	1班	计算机科学与技术	18966600016	河南新乡
4z081107	李艳	1班	计算机科学与技术	18966700017	陕西西安
4z081201	叶俊杰	2班	计算机应用技术	18966800018	陕西西安
4z081202	柳俊杰	2班	计算机应用技术	18966600019	陕西西安
4z081203	王鑫	2班	计算机应用技术	18966400021	陕西西安

记录: 10 共有记录数: 11

图 9-3 数据表

2. Access 支持的数据类型(如表 9-1 所示)

1) 文本类型

文本类型用于存储文本或文本与数字组合的数据。Access 默认文本型字段大小是 50 个字符,字符文本型字段最长为 255 个字符。设置“字段大小”属性可控制可输入的最大字符长度。

2) 备注类型

备注类型用于存储长文本及数字,例如注解或者说明等。备注型可以解决文本型无法解决的问题,如果取值的字符个数超过 255 个字符,就无法使用文本类型,而可以使用备注型。备注型允许存储的内容最长为 64 000 个字符。

3) 数字类型

数字类型用于存储可以进行算术运算的数据,有关货币的计算除外。用户可以通过设置“字段大小”属性定义一个特定的数据类型。

4) 日期/时间类型

日期/时间类型用于存储日期或时间数据。每个日期/时间数据类型字段需要 9 个字节的存储空间。

5) 货币类型

货币类型用于存储货币值,在计算时禁止四舍五入。精确度为整数部分 15 位数,小数部分 4 位数,货币数据类型长度为 8 个字节。

6) 自动编号类型

自动编号类型比较特殊。每次向表中添加新记录时,Access 会自动插入唯一顺序号,即在自动编号字段中指定数值。自动编号数据类型长度为 4 个字节。

7) 是/否类型

是/否数据类型适合存储只有两种可能性的数据,如是/否、真/假,它不允许 NULL 值,比文本数据类型节省存储空间。

8) OLE 对象类型

OLE 对象类型用于存储声音、图片、图像等多媒体数据,或是 Word、Excel 文件,或是用其他应用软件创建的数据。OLE 对象类型是指字段允许单独地“链接”或“嵌入”OLE 对

象。必须在窗体或报表中使用结合对象框来显示 OLE 对象。OLE 对象类型最大可为 1GB(受磁盘空间限制)。

表 9-1 Access 的数据类型

数据类型	用　途	字符长度
文本	字符和数字	0～255 个字符
备注	字符和数字	0～64 000 个字符
数字	数值	1、2、4 或 8 字节
日期/时间	日期/时间	8 字节
货币	数值	8 字节
自动编号	自动数字	4 字节
是/否	是/否、真/假	1 字节
OLE 对象	链接或嵌入对象	可达 1GB
超链接	Web 地址、邮件地址	可达 64 000 字节
查阅向导	来自其他表或列表的值	通常为 4 字节

对于某一具体数据而言，可以使用的数据类型可能有多种，例如电话号码可以使用数字型，也可使用文本型，但只有一种是最合适的。

在使用数据类型时主要考虑以下几个方面：

- 字段中可以使用什么类型的值。
- 需要用多少存储空间来保存字段的值。
- 是否需要对数据进行计算(主要区分是用数字，还是文本、备注等)。
- 是否需要建立排序或索引(备注、超链接及 OLE 对象型字段不能使用排序和索引)。
- 是否需要进行排序(数字和文本的排序有区别)。
- 是否需要在查询或报表中对记录进行分组(备注、超链接及 OLE 对象型字段不能用于分组记录)。

9.1.3 设定数据表之间的关系

数据库中的各表之间并不是孤立的，它们彼此之间存在或多或少的联系，这就是“表间关系”。可以通过表之间的联系来访问每个表中的数据，实现数据共享。

1. 表的索引

当表中的数据很多时，需要利用索引帮助用户更有效地查询数据。

索引的概念涉及记录的物理顺序与逻辑顺序。文件中的记录一般按其磁盘存储顺序输出，这种顺序称为物理顺序。索引不改变文件中记录的物理顺序，而是按某个索引关键字(或表达式)来建立记录的逻辑顺序。在索引文件中，所有关键字值按升序或降序排列，每个值对应原文件中相应的记录的记录号，这样便确定了记录的逻辑顺序。今后的某些对文件记录的操作可以依据这个索引建立的逻辑顺序来操作。

如图 9-4 所示，第一张表是原表文件内容，第二张表是依据“学生姓名”建立的一个排序文件，第三张表是依据“学生姓名”建立的一个索引文件。

显然，索引文件也会增加系统开销，我们一般只对需要频繁查询或排序的字段创建索

Record#	学生姓名	指导老师	老师电话
1	刘小景	钱志国	5666043
2	李 娟	杨一如	5666120
3	古介新	吴 萌	4108219
4	屈 达	钱志国	5666043
5	王成义	吴 萌	4108219

Record#	学生姓名	指导老师	老师电话
1	Lij李 娟	杨一如	5666120
2	Liu刘小景	钱志国	5666043
3	Guj古介新	吴 萌	4108219
4	Qud屈 达	钱志国	5666043
5	Wan王成义	吴 萌	4108219

Record#	学生姓名
2	Lij李 娟
1	Liu刘小景
3	Guj古介新
4	Qud屈 达
5	Wan王成义

图 9-4　原表文件和索引文件

引。而且，如果字段中许多值是相同的，索引不会显著提高查询效率。

字段数据类型为文本、数字、货币、日期/时间型的字段值能进行索引设置，搜索保存在字段中的值，排序字段中的值。

表的主键将自动被设置为索引，而备注、超链接及 OLE 对象等类型的字段则不能设置索引。

Access 2003 为每个字段提供了 3 个索引选项：无、有(有重复)、有(无重复)。

索引可分为单一字段索引和多字段索引两种。一般情况下，表中的索引为单一字段索引。建立单一字段索引的方法如下(如图 9-5 所示)：

(1) 打开表设计视图，单击要创建索引的字段，该字段属性将出现在“字段属性”区域中。

(2) 打开“常规”选项卡的“索引”下拉列表，在其中选择“有(有重复)”选项或“有(无重复)”选项即可。

(3) 保存修改。



图 9-5　建立索引

如果经常需要同时搜索或排序更多的字段，那么就需要为组合字段设置索引。建立多字段索引的操作步骤如下：

（1）在表的设计视图中单击工具栏中的“索引”按钮，弹出“索引”对话框。

（2）在“索引名称”列的第一个空行内输入索引名称，索引名称一般与索引字段名相同。

（3）选择字段名称，设置排序次序。

注意：在建立表之间的关联时，首先需要建立索引，然后才能进行查询。

2. 表的主关键字

数据库中的每一个表都必须有一个主关键字。它用于保证表中的每条记录都是唯一的。定义主键的方法很简单，在保存表时(该表未定义主键)，Microsoft Access 会弹出消息对话框，询问用户是否创建主键。如果选择“否”，则不创建主键；选择“是”，Access 会自动创建一个自动编号类型的字段并添加到表的第一列，作为该表的主键。更改主键时，首先要删除旧的主键。而删除旧的主键，先要删除其被引用的关系。

3. 创建并查看表间关系

可以在包含类似信息或字段的表之间建立关系。在表中的字段之间可以建立 3 种类型的关系：一对一、一对多、多对多。

一对一关系存在于两个表中含有相同信息的相同字段，即一个表中的每条记录都只对应于相关表中的一条匹配记录。如雇员表和人力资源表。

一对多关系存在于当一个表中的每一条记录都对应于相关表中的一条或多条匹配记录时，如产品表与销售表。

在表与表之间建立关系，不仅确立了数据表之间的关联，还确定了数据库的参照完整性。即在设定了关系后，用户不能随意更改建立关联的字段。参照完整性要求关系中一张表中的记录在关系的另一张表中有一条或多条相对应的记录。

不同的表之间的关联是通过表的主键来确定的，因此当数据表的主键更改时，Access 2003 会进行检查。

创建数据库表关系的方法如下：

（1）单击数据库窗口工具栏上的“关系”按钮，或者选择“工具”→“关系”命令，打开关系窗口。选择“显示表”，将表添加到设计窗口中，如图 9-6 所示。

（2）拖动一个表的主键到关联表的相应字段处，如图 9-7 所示。根据要求重复此步骤。



图 9-6　添加表

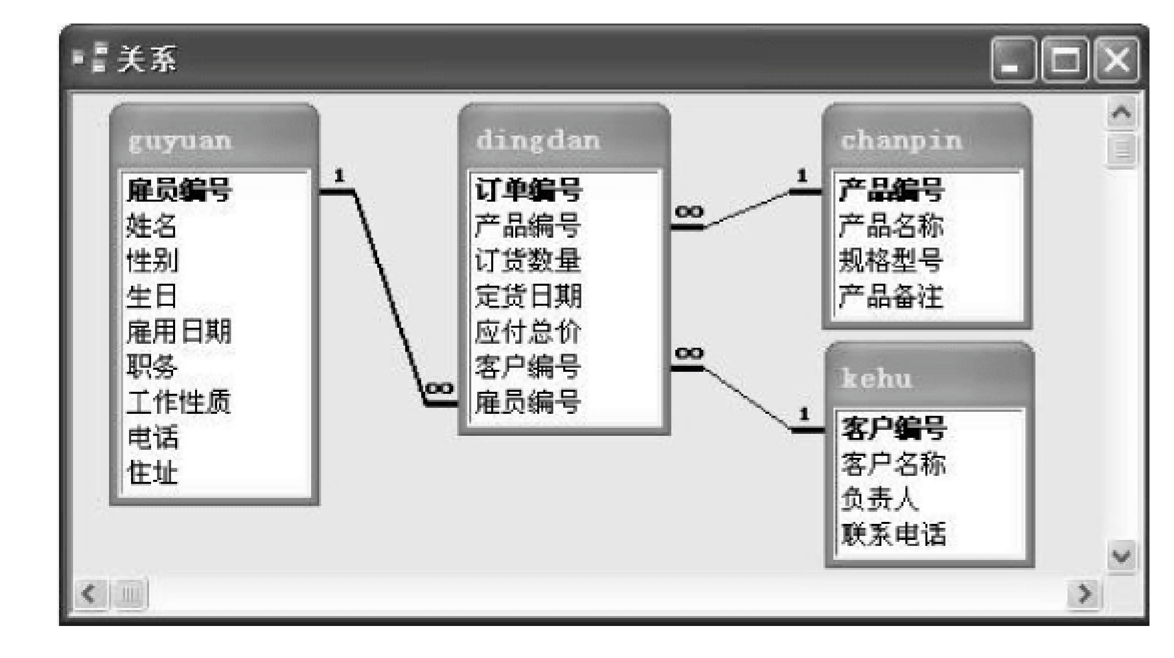


图 9-7　添加表关系

关系可以查看和编辑。打开“关系”窗口，即可查看关系；而双击表间的连线，可以编辑任何连接关系，此时弹出编辑窗口，如图 9-8 所示。



图 9-8 编辑关系

4. 表间关系的编辑与删除

用户可以编辑已有的关系，或删除不需要的关系。如图 9-9 所示，双击关系连线，可编辑关系；而右击连线，选择“删除”，可删除关系。

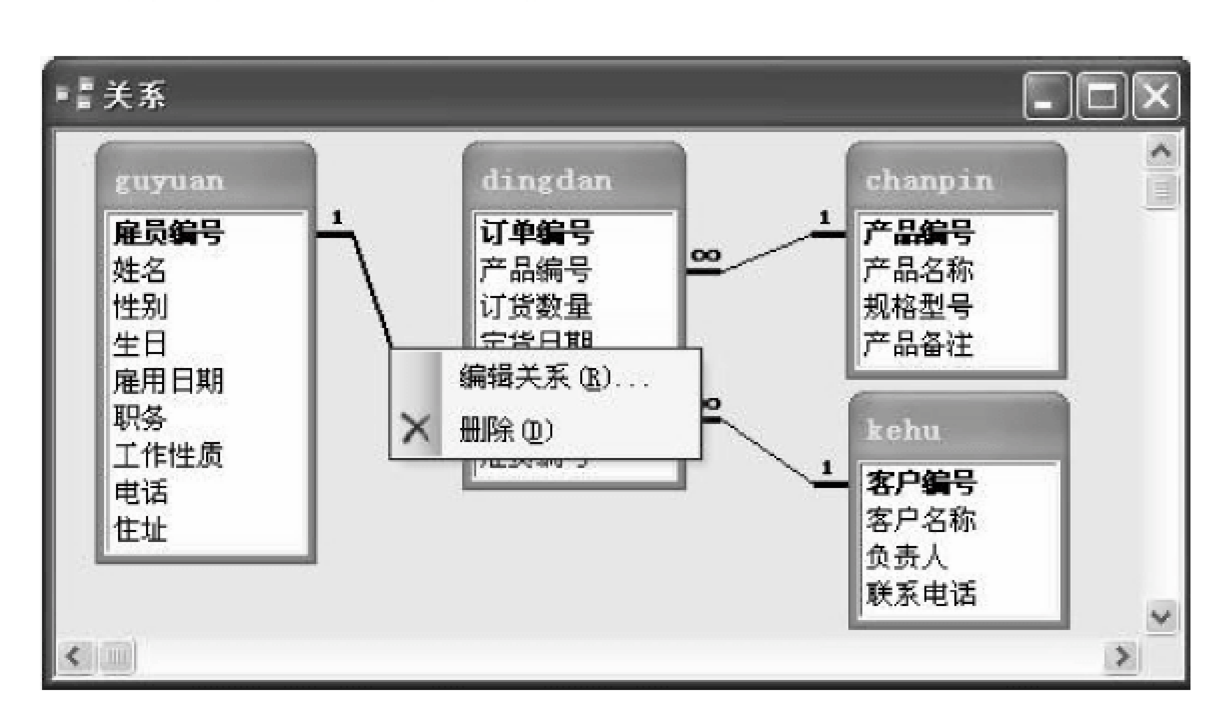


图 9-9 删除关系

9.1.4 数据表中的数据操作

1. 更改数据表的显示方式

用户可根据需要来选择不同的字体。选择“格式”→“字体”命令，将弹出“字体”对话框。

用户可以对数据表的单元格效果进行设置。其操作方法为：选择“格式”→“数据表”命令，弹出“设置数据表格式”对话框。

2. 修改数据表中的数据(如图 9-10 所示)

1) 插入新数据

当向一个空表或者向已有数据的表增加新的数据时，都要使用插入新记录的功能。

2) 修改数据

在数据表视图中，用户可以方便地修改已有的数据记录。注意保存。



图 9-10 插入、修改、删除数据

3）替换数据

如果想把数据表中的某个数据替换为另一个数据，可以进行如下操作：在数据表视图中选中要替换的字段内容，然后选择“编辑”→“替换”命令，弹出“查找和替换”对话框。

4）复制、移动数据

利用剪贴板功能可以很方便地进行复制、移动数据操作。

5）删除记录

可以利用“编辑”→“删除”进行删除操作，也可用快捷键方式完成该操作。

3. 排列数据

Access 根据主键值自动排序记录。在数据检索和显示期间，用户可以按不同的顺序来排序记录。在数据表视图中，可以对一个或多个字段进行排序。升序的规则是按字母顺序排列文本，从最早到最晚排列日期/时间值，从最低到最高排列数字与货币值，如图 9-11 所示。

对于多个字段的排序，Access 2003 使用从左到右的优先权排序。

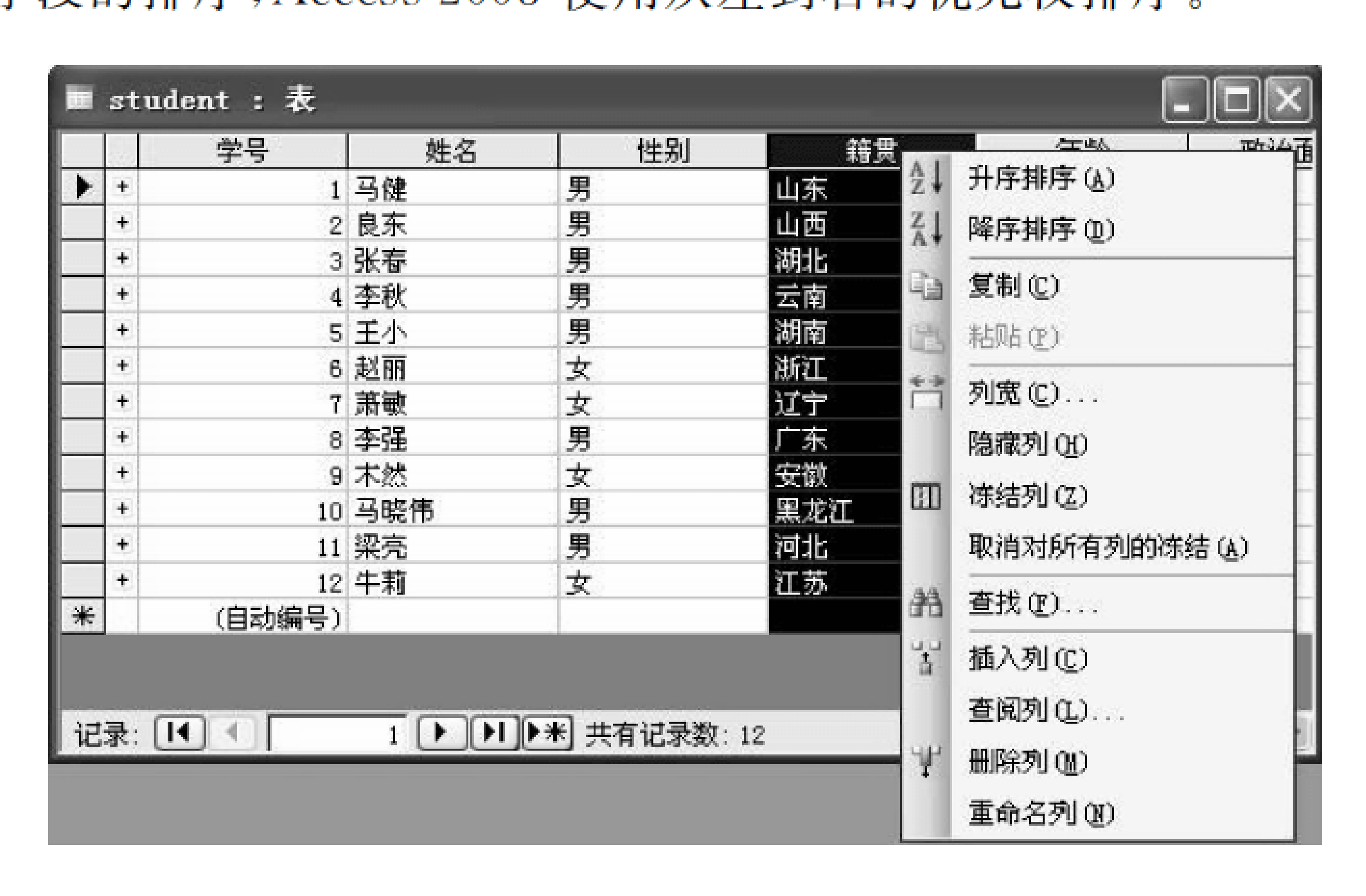


图 9-11 排列数据

4. 查找数据

用户可以在数据表视图中查找指定的数据，其操作是通过"编辑"→"查找"命令来完成的。

5. 筛选数据

筛选数据是只将符合筛选条件的数据记录显示出来，以便用户查看。筛选方法有5种，分别为按窗体筛选、按选定内容筛选、输入筛选、高级筛选/排序、内容排除筛选，如图9-12所示。

图 9-12 数据筛选

6. 高级筛选与排序

高级筛选与排序可以应用于一个或多个字段的排序或筛选。高级筛选/排序窗口分为上下两部分，上面是含有表的字段列表，下面是设计网格，如图9-13所示。

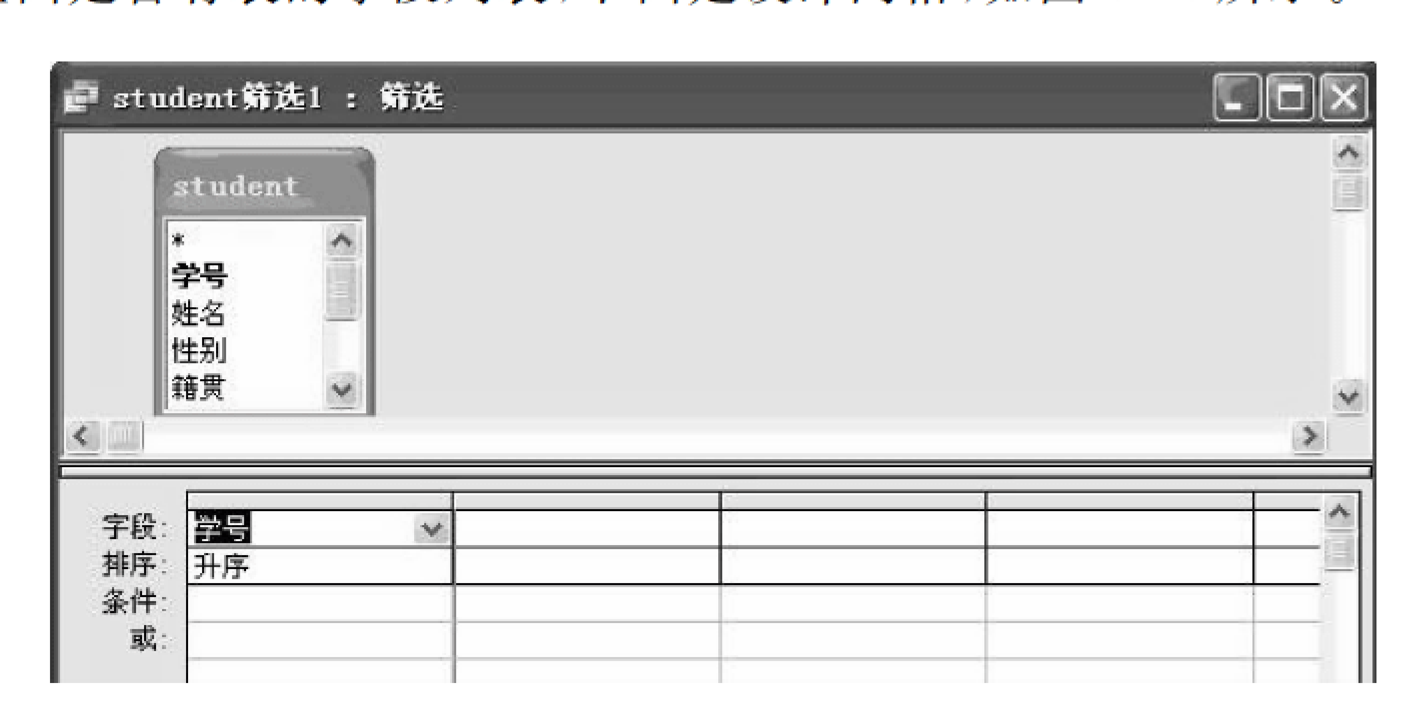


图 9-13 高级数据筛选

要创建一个高级筛选，首先要把字段添加到用于排序和规定筛选准则的设计网格中。在"条件"行中，可添加要显示记录的条件，它的设置方法与按窗体筛选的设置方法一样。

用户如果保存了筛选，则该筛选与表一起保存，而不作为独立的对象保存。当用户再次打开该表时，筛选不再起作用。如果用户想在一个表中使用多个筛选或永久保存一个筛选，必须将其作为一个查询保存起来。

用户还可以取消和删除筛选，单击工具栏上的取消筛选/排序按钮即可。

若要完全删除一个筛选，就要通过“清除网格”、“应用筛选”、“关闭”、“高级筛选/排序”等操作来完成。

项目案例 1：配置 Microsoft Access 数据库文件的 DSN 案例。

ODBC 数据源管理器提供 3 种不同类型的 DSN：用户 DSN、系统 DSN 和文件 DSN。

用户 DSN 在 Windows NT 系统注册表中存储如何与指定数据提供者连接的信息，用户 DSN 只对当前用户可见，而且只能用于当前机器上。

系统 DSN 也在系统注册表中存储数据库的信息，系统 DSN 添加的数据源本机上的所有用户都可以看到。

文件 DSN 是一个文本文件，它包含数据库的连接信息，该文件保存在计算机中，而且可以通过复制轻易地从一个服务器转移到另一个服务器。在 Windows 系统下，打开“控制面板”窗口，就可以创建基于 DSN 的文件了。

创建 ODBC 数据源的步骤如下：

(1) 打开“控制面板”→“管理工具”→“数据源(ODBC)”，显示“ODBC 数据源管理器”对话框，如图 9-14 所示。

(2) 单击“添加”按钮，显示“创建新数据源”对话框。选择 Microsoft Access Driver(＊.mdb)，单击“完成”按钮，进入下一步。

(3) 给数据源命名(如 MyData)，再单击“选择(S)...”，选择 Access 数据库，单击“确定”按钮，就 OK 了。



图 9-14 文件 DSN 配置

9.2 SQL Server 数据库

SQL Server 是由美国 Microsoft 公司推出的高性能关系数据库管理系统，它经历了十多年的发展过程，目前已经成为很出色的数据库管理系统。

9.2.1　安装 SQL Server 服务器

将 SQL Server 2000 的光盘插入 CD-ROM 驱动器之后，SQL Server 的安装程序会自动启动。如果该光盘不自动运行，请双击该光盘根目录中的 Autorun.exe 文件，打开安装选项窗口，如图 9-15 所示。



图 9-15　SQL Server 安装界面

操作步骤如下：

第一步：将企业版安装光盘插入光驱(以常用的简体中文 4IN1 版本为例)，选择好版本后即可见如图 9-16 所示窗口，选择“安装 SQL Server 2000 组件”，弹出如图 9-17 所示窗口，选择“安装数据库服务器”。

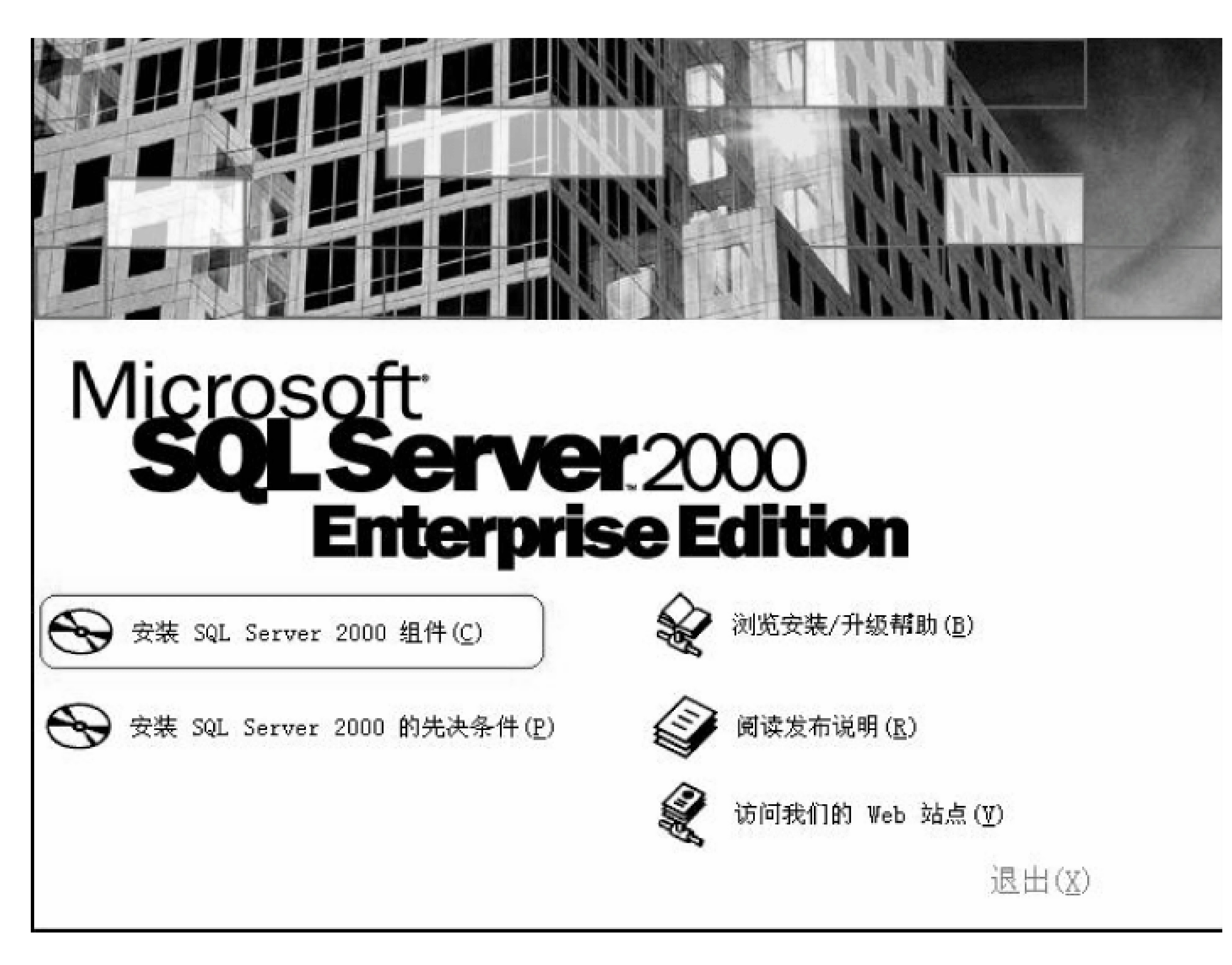


图 9-16　安装 SQL Server 组件



图 9-17 安装数据库服务器

注意：*系统在某些版本上会出现提示信息，意思是说安装好 SQL Server 2000 服务器后需要安装 SP4 安装补丁来更新数据库服务器。*

第二步：根据 SQL Server 安装向导进行下一步安装。当遇到确认计算机名时，选择“本地计算机”，如图 9-18 所示。在“安装选择”窗口选择“创建新的 SQL Server 实例……”。对于初次安装的用户，应选用这一安装模式，无须使用“高级选项”进行安装。“高级选项”中的内容均可在安装完成后进行调整。在“用户信息”窗口输入用户信息，单击“下一步”按钮并接受软件许可证协议，单击“是”按钮。



图 9-18 计算机名

第三步：在“安装定义”窗口选择“服务器和客户端工具”选项进行安装，如图 9-19 所示。我们需要将服务器和客户端同时安装，这样在同一台机器上可以完成相关的所有操作，

对于我们学习 SQL Server 很有用处。如果已经在其他机器上安装了 SQL Server，则可以只安装客户端工具，用于对其他机器上 SQL Server 的存取。



图 9-19　安装定义

在“实例名”窗口选择“默认”的实例名称。这时 SQL Server 的名称将和 Windows 2000 服务器的名称相同。例如，Windows 服务器名称是 mysql，则 SQL Server 的名字也是 mysql。SQL Server 2000 可以在同一台服务器上安装多个实例，也就是可以重复安装几次。这时就需要选择不同的实例名称了。建议将实例名称限制在 10 个字符之内。实例名称会出现在各种 SQL Server 和系统工具的用户界面中，因此名称越短越容易读取。另外，实例名称不能是 Default 或 SQL Server 以及 SQL Server 的保留关键字等。

第四步：在“安装类型”窗口选择“典型”安装选项，并指定“目的文件夹”，如图 9-20 所示。程序和数据文件的默认安装位置都是“C:\Program Files\Microsoft SQL Server\”。一般因为 C 盘是系统区，所以选择其他盘。注意，如果数据库数据有 10 万条以上的话，请预留至少 1GB 的存储空间，以应付需求庞大的日志空间和索引空间。

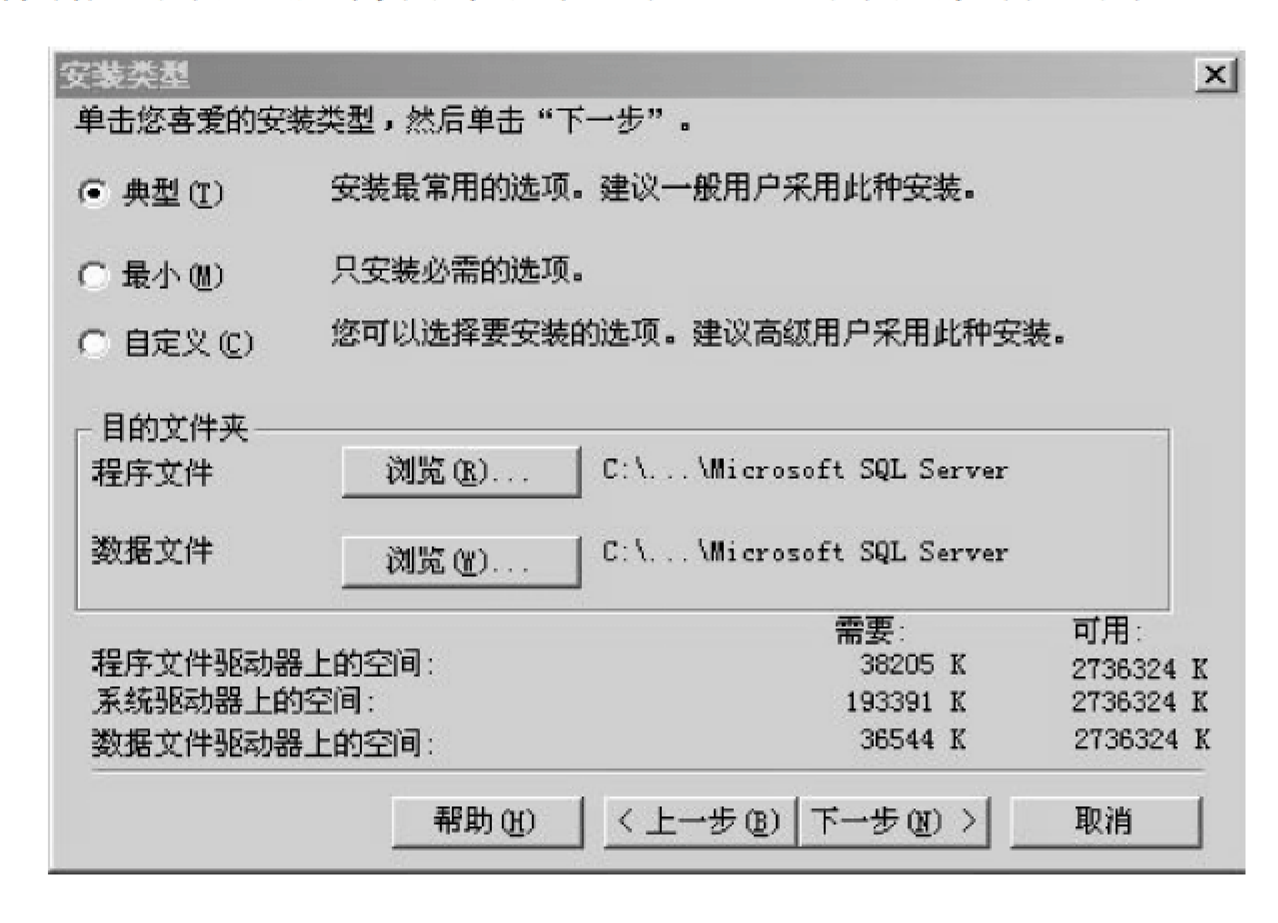


图 9-20　安装类型

在这里建议把数据库文件安装到D盘或其他非系统分区中，以减小由于系统故障带来的数据安全威胁。

在“服务账号”窗口选择“对每个服务使用统一账户……”选项。在“服务设置”处选择“使用本地系统账户”。如果需要“使用域用户账户”，请将该用户添加至 Windows Server 的本机管理员组中。

第五步：在“身份验证模式”窗口选择“混合模式……”选项，并设置管理员 sa 账号的密码”，如图 9-21 所示。如果你的目的只是为了学习的话，可以将该密码设置为空，以方便登录。

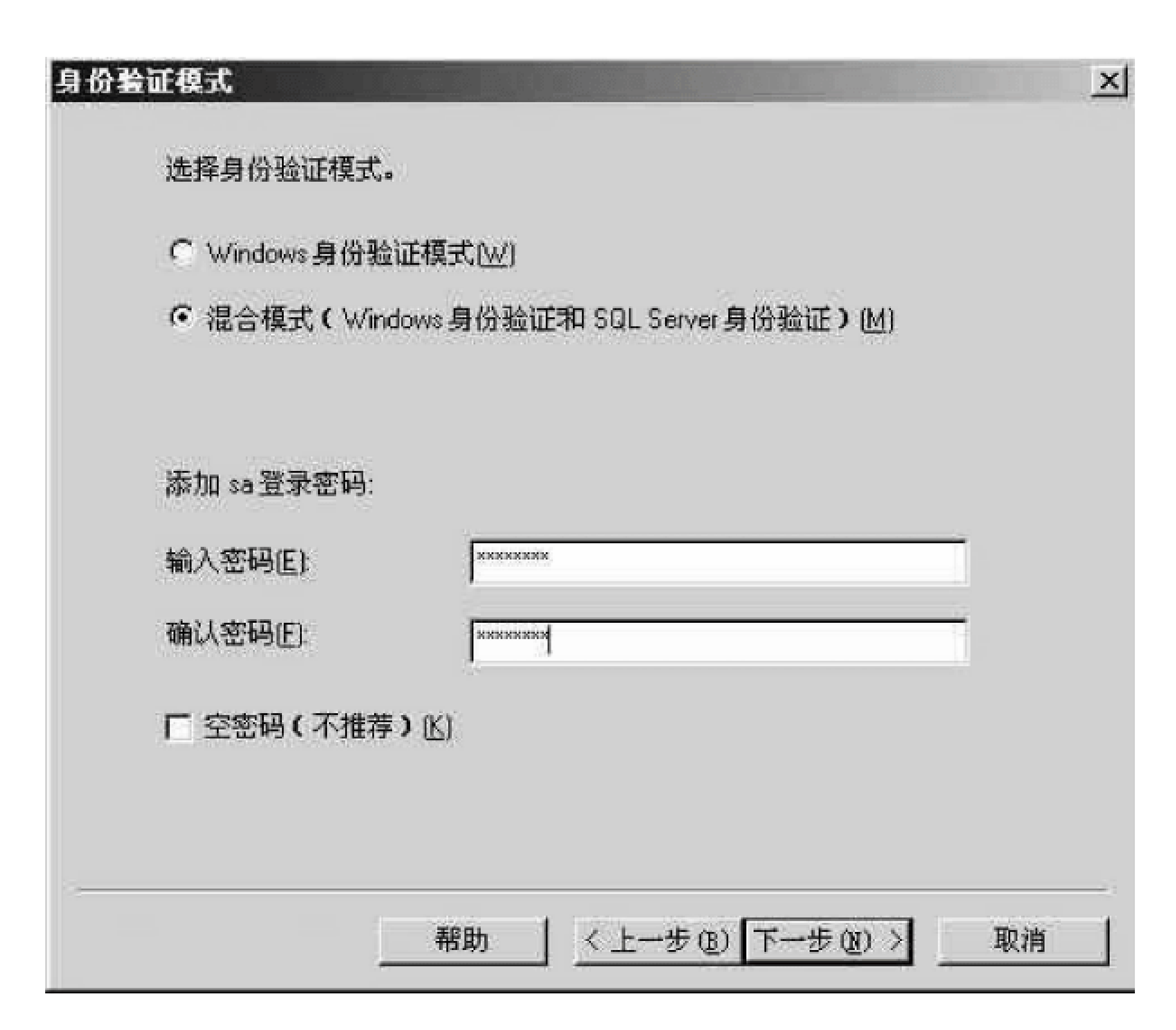


图 9-21 身份验证模式

9.2.2 启动、暂停和停止 SQL Server 服务器

在 SQL Server 2000 安装成功以后，选择“开始”|“程序”，将会出现 Microsoft SQL Server 程序组子项目，它们分别是查询分析器、导入和导出数据、服务管理器、服务器网络实用工具、客户端网络实用工具、联机丛书、企业管理器和事件探查器等。其中最常用的是服务管理器、企业管理器和查询分析器。

系统安装之后自动生成 6 个数据库，其中包含 4 个系统数据库，分别是 Master、Model、Msdb 和 Tempdb，两个示例数据库，分别是 Pubs、Northwind。

查看 SQL Server 2000 启动情况。成功安装了 SQL Server 2000，在 SQL Server 正常启动后，计算机桌面右下角出现的 SQL Server 服务监视图标显示为一个带绿色三角的服务启动标记，如图 9-22 所示。

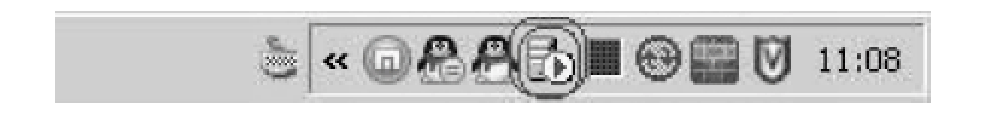


图 9-22 服务器运行

若该绿色图标变成了红色的方块，则表明 SQL Server 已停止服务，可通过“SQL Server 服务管理器”程序来启动 SQL Server。双击该图标，将出现如图 9-23 所示的“SQL Server 服务管理器”程序，可通过该程序停止、启动 SQL Server 的后台服务。

如果在计算机桌面右下角没有出现 SQL Server 服务监视图标，选择“开始”→“程序”，即可看到 Microsoft SQL Server 2000 的程序组件，单击其中的“服务管理器”，则可以启动 SQL Server 服务。



图 9-23 启动、暂停和停止 SQL Server 服务器

9.2.3 安装 SQL Server 2000 常见问题

SQL Server 安装时经常会遇到一个问题，即出现错误提示：“以前的某个程序安装已在安装计算机上创建挂起的文件操作。运行安装程序之前必须重新启动计算机。”如图 9-24 所示。



图 9-24 挂起文件操作

操作方法：打开注册表编辑器，选择“开始”→“运行”，在命令窗口中输入“regedit”。打开注册表后，在 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager 中找到 PendingFileRenameOperations 项目，并删除它。这样就可以清除挂起项目。

9.3 SQL 语句

SQL 语言是关系数据库操作语言，在过去的几年中，SQL 语言已经发展成为标准的计算机数据库语言。

1986 年 ANSI(American National Standards Institute，美国国家标准协会)和 ISO (International Standards Organization，国际标准化组织)颁布了 SQL 正式标准，同时确认 SQL 语言为数据库操作的标准语言。现在已有 100 多种遍布在从微机到大型机上的 SQL 数据库产品。SQL 语言基本上独立于数据库本身及其使用的计算机、网络、操作系统，基于 SQL 的 DBMS 开发商所提供的产品一般都具有良好的可移植性。

SQL 语言最初由 IBM 的研究人员在 20 世纪 70 年代提出，最初的名称为 SEQUEL(结果)，从 80 年代开始改名为 SQL，看似是 SEQUEL 的缩写，但一般又理解为结构化查询语言(structure query language)。

9.3.1 创建数据库、表语句

```
CREATE DATABASE LOGIN;
CREATE TABLE CJ FREE( xh C(8), kcdh C(4), cj N )
```

注意：若无 FREE 选项，将在打开的数据库中创建数据库表 CJ。

附表：学生表，如表 9-2 所示。成绩表，如表 9-3 所示。课程表，如表 9-4 所示。

表 9-2 学生表 XS. DBF 结构

xh	C	8	学号
xm	C	6	姓名
xb	C	2	性别
zydh	C	6	专业代号
ximing	C	18	系科名称
jg	C	10	籍贯
csrq	D		出生日期
zp	G		照片

表 9-3 成绩表 CJ. DBF 结构

xh	C	8	学号
kcdh	C	4	课程代号
cj	N	3	成绩

表 9-4 课程表 KC. DBF 结构

kcdh	C	4	课程代号
kcm	C	18	课程名
kss	N	2	课时数
bxk	L		必修课
xf	N	1	学分

9.3.2 查询语句

SQL 的核心是查询功能，SQL 的查询命令也称做 SELECT 命令，它的常用语法格式如下：

```
SELECT  [ALL|DISTINCT] [TOP (表达式)] <目标列表达式>        //说明要查询的数据
FROM [数据库名!]<表名>                                      //说明数据来源
[[INNER|LEFT[OUTER]|RIGHT[OUTER]|FULL[OUTER]]              //说明与其他表的联接方式
          JOIN 数据库名!表名 ON <联接条件>]
WHERE      <条件表达式>                                      //说明查询的条件
[GROUP BY <分组依据>]                                        //对查询结果进行分组
[HAVING <条件表达式>]                                        //限定分组满足的条件
[ORDER BY <排序依据>]                                        //对查询结果进行排序
[UNION [ALL]...]                                             //对多个查询结果进行合并
```

以上语句是必须要掌握的 SQL SELECT 命令。

1. 简单查询

简单查询有两种：一种是由 SELECT 和 FROM 短语构成的无条件查询；另一种是由 SELECT 和 FROM 以及 WHERE 短语构成的条件查询。

【例 9-1】 查询学生表中所有字段。

```
SELECT * FROM XS
```

注意："*"是通配符，代表全部字段列表。

【例 9-2】 查询学生表中所有学号和姓名。

```
SELECT xh ,xm FROM XS
```

注意：字段名之间要用英文逗号分隔。

【例 9-3】 从学生表中查询所有系科名称，要求不重复显示相同的系科名称。

```
SELECT DISTINCT ximing  FROM XS
```

注意：如果要去掉重复值，只须使用 DISTINCT 短语。

【例 9-4】 从成绩表中查询所有成绩大于 85 分的学号。

```
SELECT DISTINCT xh  FROM cj WHERE CJ.cj>85
```

注意：简单条件查询要使用 WHERE 短语。

2. 简单联接查询

联接是关系的基本操作之一，简单联接查询是一种基于多个关系(往往是基于多张表)的查询。

【例 9-5】 查询至少有一门课程成绩大于 85 的学生姓名。

```
SELECT xm FROM XS,CJ WHERE CJ.cj>85 and XS.xh=CJ.xh
```

注意：

(1) 这里所要查询的数据分别来自"学生表"和"成绩表"。

(2) 如果在 FROM 之后有两个表，那么这两个表之间一定有一种关系，如本例中的 XS 表和 CJ 表都有"学号"字段，否则无法构成检索表达式。

(3) 本例中"XS. xh=CJ. xh"是联接条件。

(4) 当 FROM 后面有多个关系含有相同的字段名时,必须用表别名前缀直接指明字段所属的表,如"XS. xh"。

注意:在 SQL 语句的 WHERE 子句后的联接条件,除了使用 VFP 的关系表达式和逻辑表达式外,还可以使用几个特殊运算符:[NOT]IN、[NOT]BETWEEN... AND...、[NOT]LIKE。

【例 9-6】 在学生表中查询所有计算机系的学生,不要其他的学生信息。

```
SELECT * FROM XS WHERE ximing LIKE "计算机系"
```

注意:可以使用 NOT 运算符来设计否定条件,如查询所有非计算机系的学生。

```
SELECT * FROM XS WHERE NOT (ximing LIKE "计算机系")
```

【例 9-7】 在学生表中查询所有姓李的学生。

```
SELECT * FROM XS WHERE xm LIKE "李%"
```

注意:"%"匹配多个任意符号,"_"匹配一个任意符号。

【例 9-8】 查询所有成绩在 80 和 90 之间的学生。

```
SELECT XS.xm FROM XS,CJ WHERE (CJ.cj BETWEEN 80 AND 90) AND (XS.xh = CJ.xh)
```

3. 计算查询

【例 9-9】 计算英语课程的平均成绩、最高成绩、最低成绩。

```
SELECT kcm,AVG(CJ.cj) as "平均成绩", MAX(CJ.cj) as "最高成绩",;
    MIN(CJ.cj) as "最低成绩" FROM CJ,KC ;
    WHERE kcm = "英语" and CJ.kcdh = KC.kcdh
```

【例 9-10】 统计学生表中的专业数目。

```
SELECT COUNT( DISTINCT zydh ) FROM XS
```

注意:COUNT 函数的参数可以是字段名,也可用"*"号来通配,如 COUNT(*)。加 DISTINCT 是为了排除相同的项。

4. 分组查询

SQL 不仅具有一般的检索功能,还可直接对检索结果进行计算,用于计算的函数有 COUNT()——计数、SUM()——求和、AVG()——求平均值、MAX()——求最大值、MIN()——求最小值。

SQL 提供的计算函数可以对满足条件的记录进行各种计算,这些函数还可以从一组值中计算出一个汇总信息,通常和 GROUP BY 分组子句配合使用,完成特定的查询。

【例 9-11】 统计每门课程的名称、平均成绩。

```
SELECT KC.kcm,AVG(CJ.cj) as "平均成绩" FROM KC,CJ ;
    GROUP BY CJ.kcdh WHERE CJ.kcdh = KC.kcdh
```

【例 9-12】 查询最少选修 3 门课程的学生姓名。

```
SELECT XS.xm FROM XS WHERE xh IN ;
    (select xh from CJ group by xh HAVING count( * )>= 3)
```

注意：HAVING 子句的作用是指定查询的结果所满足的条件，通常和 GROUP BY 配合使用；而 WHERE 子句的作用是指定参与查询的表中的数据所满足的条件。

【例 9-13】 按系科统计各专业男生人数、女生人数、系科名称、性别。

```
SELECT XS.ximing,XS.xb,COUNT(XS. * ) as "学生人数" FROM XS ;
    GROUP BY XS.ximing , XS.zydh , XS.xb
```

注意：题目要求按系科、专业、性别进行统计，因此应按系科、专业、性别来分组；分组表达式若有多个，用逗号分隔。

5. 排序

SQL 使用 ORDER BY 进行排序的操作，默认是升序排序。

【例 9-14】 按学号升序、成绩降序检索学生成绩。

```
SELECT * FROM CJ ORDER BY xh ASC , cj DESC
```

6. 查询输出去向

FROM 子句中的 INTO 与 TO 选项用于指定查询结果的输出去向，默认为浏览窗口。INTO 选项有 3 种输出去向：ARRAY <数组>、CURSOR <临时表>、TABLE|DBF <表名>；TO 选项也有 3 种输出去向：文件、屏幕、打印机。

【例 9-15】 查询学生名单，并生成永久表 STUDENT.DBF。

```
SELECT * FROM XS INTO DBF STUDENT
```

【例 9-16】 查询前 10 名学生名单，将查询结果保存在文本文件 TMP.TXT 中。

```
SELECT TOP 10 * FROM XS TO FILE TMP ADDITIVE
```

注意：TOP 10 表示只取前 10 条记录。ADDITIVE 选项表示以追加方式在文件尾部添加数据，否则为覆盖方式。

【例 9-17】 查询前 10%的学生名单，将查询结果保存在临时表 TEMP 中。

```
SELECT TOP 10 PERCENT * FROM XS INTO CURSOR TEMP
```

注意：TOP 10 PERCENT 表示只取全部记录的前 10%条记录。

【例 9-18】 查询学生名单，将查询结果保存在数组 A 中。

```
SELECT * FROM XS INTO ARRAY A
```

9.3.3 插入语句

在一个表的尾部追加记录时，使用插入操作。用 SQL 语言插入记录的语法格式如下：

```
INSERT INTO <表名>[(字段名列表)] VALUES(<表达式 1[,表达式 2… ]>)
```

【例 9-19】 在学生表中插入一条新记录。

```
INSERT INTO XS(xh , xm , xb , csrq);
    VALUES('01001023', '王平', '男',{^ 1984 - 05 - 03} )
```

9.3.4 修改语句

用 SQL 语言修改记录的语法格式如下：

```
UPDATE [!数据库名]<表名> SET <字段名 1 > = <表达式 1 >[,<字段名 2 > = <表达式 2 >...];
    WHERE <条件表达式>
```

【例 9-20】 将成绩表中所有课程代号为 2 的分数降低 2%。

```
UPDATE CJ SET cj = 0.98 * cj WHERE kcdh = '2'
```

注意：若省略 WHERE 子句，将对表中全部记录进行修改。

9.3.5 删除语句

用 SQL 语言删除记录的语法格式如下：

```
DELETE FROM <表名> WHERE <条件表达式>
```

【例 9-21】 删除学生表中“王平”的记录。

```
DELETE FROM XS WHERE xm =  '王平'
```

注意：DELETE 的结果只是逻辑删除记录，只有执行 PACK 才能物理删除。

项目案例 2：使用 SQL 语句创建数据库代码案例。

【题目】 制作在线考试网站后台数据库创建代码，包括表设计、记录的插入等。

【代码】

```
CREATE DATABASE stuDB
USE stuDB                                        //将当前数据库设置为 stuDB
GO
CREATE  TABLE  stuInfo                           /* - 创建学员信息表 - */
(
 stuName  VARCHAR(20)  NOT  NULL ,               //姓名,非空(必填)
 stuNo  CHAR(6)  NOT  NULL,                      //学号,非空(必填)
 stuAge  INT  NOT  NULL,                         //年龄,INT 类型默认为 4 个字节
 stuID  NUMERIC(18,0),                           //身份证号
 stuSeat  SMALLINT  IDENTITY (1,1),              //座位号,自动编号
 stuAddress  TEXT                                //住址,允许为空,即可选输入
)
GO
CREATE TABLE stuMarks
(
 ExamNo  CHAR(7)  NOT NULL,                      //考号
```

```
 stuNo  CHAR(6)  NOT NULL,                         //学号
 writtenExam  INT  NOT NULL,                       //笔试成绩
 LabExam  INT  NOT NULL                            //机试成绩
)
GO
USE stuDB
GO
SELECT 考号 = ExamNo,学号 = stuNo,笔试 = writtenExam,机试 = labExam,
  平均分 = (writtenExam + labExam)/2,
  等级 = CASE
         WHEN (writtenExam + labExam)/2 < 60 THEN '不及格'
         WHEN (writtenExam + labExam)/2 BETWEEN 60 AND 69 THEN '差'
         WHEN (writtenExam + labExam)/2 BETWEEN 70 AND 79 THEN '中'
         WHEN (writtenExam + labExam)/2 BETWEEN 80 AND 89 THEN '良'
         ElSE '优'
       END
  FROM stuMarks
```

9.4 备份和恢复数据库

当使用一个数据库时，总希望数据库的内容是可靠的、正确的，但由于计算机系统的故障(硬件故障、软件故障、网络故障、进程故障)影响数据库系统的操作，影响数据库中数据的正确性，甚至破坏数据库，使数据库中全部或部分数据丢失，因此当发生上述故障后，希望能重构这个完整的数据库。数据库排除故障的通常做法是备份和恢复数据库。

1. 备份数据库

备份数据库是将数据库中的数据库对象备份到数据库以外的地方。备份分为两种：物理备份和逻辑备份。通常备份的内容有系统数据库、用户数据库、事务日志。

物理备份是对数据库的操作系统物理文件(如数据文件、控制文件和日志文件等)的备份。

逻辑备份是对数据库逻辑组件(如表、视图和存储过程等数据库对象)的备份。

备份的材料有硬盘、磁带等。硬盘是常用的备份方法，用于备份本地文件，也可以备份网络文件。磁带是大容量的备份方法，磁带仅用于备份本地文件。

备份时，通常考虑系统备份的时间，以便系统、数据库出现故障时，能够恢复到最近、最新的备份时间点。如果系统环境为联机事务处理，则应当经常备份数据库；如果系统只执行少量作业，则适当进行备份。

备份的步骤如下：

(1) 计算所需备份的数据量。如果是网络备份，数据量应是所需备份服务器的数据量的总和。

(2) 考虑备份窗口/恢复窗口的时间差别。如果恢复窗口和备份窗口相同，须采用全备份方法。

(3) 确定磁带所需保留的周期。

(4) 计算对比设备性能表,选择合适的备份设备。

(5) 根据实际情况,选择合适的备份软件。

2. 恢复数据库

恢复操作是与备份相对应的操作。

数据库恢复能够及时还原和重建数据库,但不是所有的情况下都能够实现恢复操作。当系统发现以下情况时,恢复操作是不能进行的:

(1) 使用与恢复的数据库名称不同的数据库名去恢复数据库。

(2) 服务器上数据库文件组与备份中的数据库文件组不一致。

(3) 须恢复的数据库名或文件名与备份的数据库名或文件名不同。

数据库恢复可以分为以下两类:

1) 实例故障的一致性恢复

当实例意外地(如掉电、后台进程故障等)或预料地(发出 SHUTDOUM ABORT 语句)中止时出现实例故障,此时需要实例恢复。实例恢复将数据库恢复到故障之前的事务一致状态。如果在当前使用的设备发现实例故障,则需要硬盘、磁盘等恢复。在下次数据库启动时(对新实例装配和打开),自动执行实例恢复。恢复的处理步骤如下:

(1) 为了解恢复数据文件中没有记录的数据,要向前滚。该数据记录在在线日志中,包括对回滚段的内容恢复。

(2) 回滚未提交的事务,按第一步重新生成回滚段所指定的操作。

(3) 释放在故障时正在处理事务所持有的资源。

(4) 解决在故障时未提交的故障事务。

2) 介质故障或文件错误的不一致恢复

介质故障是当一个文件、一个文件的部分或磁盘不能读或不能写时出现的故障。文件错误一般指意外的错误导致文件被删除或意外事故导致文件的不一致。这种状态下的数据库都是不一致的,需要数据库管理员即 DBA 手工来进行数据库的恢复。

项目案例 3:数据库恢复案例测试环境。

(1) 数据库环境。

以下的所有案例都是通过测试的,环境为:

```
OS: Windows 2000 Server
DB: SQL Server2000
DBNAME: TEST
```

数据文件:

```
SQL> select file#,status,enabled,name from v$datafile;
FILE# STATUS ENABLED NAME
-----------------------------------------------------------------
  1 SYSTEM READ WRITE D:/SQL Server2000/ORADATA/TEST/SYSTEM01.DBF
  2 ONLINE READ WRITE D:/SQL Server2000/ORADATA/TEST/RBS01.DBF
  3 ONLINE READ WRITE D:/SQL Server2000/ORADATA/TEST/USERS01.DBF
  4 ONLINE READ WRITE D:/SQL Server2000/ORADATA/TEST/TEMP01.DBF
```

```
5 ONLINE READ WRITE D:/SQL Server2000/ORADATA/TEST/TOOLS01.DBF
6 ONLINE READ WRITE D:/SQL Server2000/ORADATA/TEST/INDX01.DBF
```

控制文件：

```
SQL> select * from v$controlfile;

STATUS NAME
----------------------------------------------------------------------
D:/SQL Server2000/ORADATA/TEST/CONTROL01.CTL
D:/SQL Server2000/ORADATA/TEST/CONTROL02.CTL
D:/SQL Server2000/ORADATA/TEST/CONTROL03.CTL
```

联机日志：

```
SQL> select * from v$logfile;

GROUP# STATUS MEMBER
----------------------------------------------------------------------
1 STALE D:/SQL Server2000/ORADATA/TEST/REDO01.LOG
2 D:/SQL Server2000/ORADATA/TEST/REDO02.LOG
3 STALE D:/SQL Server2000/ORADATA/TEST/REDO03.LOG
```

(2) 数据库备份脚本。

冷备份脚本：

```
rem   script:coldbak.sql
rem   creater:chenjiping
rem   date:5.8.2003
rem   desc:offline full backup database
-- connect database
connect internal/password;
-- shutdown database
shutdown immediate;
-- Copy Data file
!xcopy d:/SQL Server2000/oradata/test/*.dbf d:/database/H/R;
-- Copy Control file
!xcopy d:/SQL Server2000/oradata/test/*.ctl d:/database/H/R;
-- Copy Log file
!xcopy d:/SQL Server2000/oradata/test/*.log d:/database/H/R;
-- startup database
startup;
```

说明：

① 以上脚本在数据库关闭状态下备份数据库所有的数据文件、联机日志、控制文件(在一个目录下)，如果成功备份，所有文件是一致的。

② 没有备份参数文件，参数文件可以另外备份，没有必要每次都备份，只须在改变设置后备份一次。

③ 如果以上命令没有成功依次执行，那么备份将是无效的。例如，连接数据库不成功，那么肯定关闭数据库也不成功，那么备份则无效。

④ 冷备份在人工干预下执行。

数据库 OS 热全备份脚本：

```
rem   script:hotbak.sql
rem   creater:chenjiping
rem   date:5.8.2003
rem   desc:backup all database datafile in archive
 -- connect database
connect internal/password;
 -- archive
alter system archive log current;
 -- start
alter tablespace system begin backup;
!xcopy d:/SQL Server2000/oradata/test/system01.dbf d:/databak/H/R;
alter tablespace system end backup;
alter tablespace rbs begin backup;
!xcopy d:/SQL Server2000/oradata/test/rbs01.dbf d:/databak/H/R;
alter tablespace rbs end backup;
alter tablespace users begin backup;
!xcopy d:/SQL Server2000/oradata/test/users01.dbf d:/databak/H/R;
alter tablespace users end backup;
alter tablespace tools begin backup;
!xcopy d:/SQL Server2000/oradata/test/tools01.dbf d:/databak/H/R;
alter tablespace tools end backup;

alter tablespace indx begin backup;
!xcopy d:/SQL Server2000/oradata/test/indx01.dbf d:/databak/H/R;
alter tablespace indx end backup;
 -- end
 -- bak control file
 -- binary
alter database backup controlfile to 'd:/databak/controlbinbak.000';
 -- ascii
alter database backup controlfile to trace;
alter system archive log current;
```

说明：

① 热备份必须在数据库归档方式下才可以运行。

② 以上脚本可以在数据库运行状态下备份数据库所有的数据文件（除了临时数据文件外），没有必要备份联机日志。

③ 归档日志至少需要一次完整备份。

④ 如果以上命令没有成功依次执行，那么备份也是无效的。

以上举例说明了数据库的恢复案例的测试环境与部分备份测试脚本，其他的备份脚本可以根据以上脚本演变而来或在案例中加以说明。

9.5 导入、导出数据表

SQL Server 的数据转换通常包括两个方面：一是把其他数据系统处理的数据导入到SQL Server 数据库中；二是将数据从 SQL Server 数据库中导出到其他应用程序中。

为了解决数据转换问题，SQL Server 系统提供了一个功能强大的组件 DTS(数据转换服务)。DTS 是一组图形工具和可编程对象，包含 DTS 导入/导出向导和 DTS 设计器。DTS 提供了在 SQL Server 与 OLE DB、开发数据库连接(ODBC)的数据源、文本格式文件或 Excel 工作表之间导入、导出和转换数据的功能。

使用 DTS 将 Access 数据转移到 SQL Server 可采取以下步骤：

(1) 在 SQL Server Manager(Enterprise Manager)的工具菜单中指向 Data Transformation Services，然后单击 Import Data。

(2) 在 Choose a Data Source(选择数据源)对话框中选择 Microsoft Access 为数据源，然后输入你的.mdb 文件名(mdb 为文件扩展名)或者选择浏览文件。

(3) 在 Choose a Destination(选择数据目标)对话框中选择 Microsoft OLE DB Provider for SQL Server，再选择数据库服务器，然后单击所需的认证模式。

(4) 在 Specify Table Copy or Query(“指定表备份或查询”)对话框中单击 Copy Tables。

(5) 在 Choose a Data Source(选择数据源)对话框中单击 Select All。

在 SQL Server 2000/2005 中除了使用 DTS 进行数据的导入/导出，还可以使用 Transact-SQL 语句进行导入/导出操作。在 Transact-SQL 语句中，主要使用 OpenDataSource 函数、OPENROWSET 函数。关于函数的详细说明，请参考 SQL 联机帮助。利用下述方法，可以十分容易地实现 SQL Server、Access、Excel 数据转换，详细说明如下。

1. SQL Server 和 Access 的数据常规导入/导出

使用 DTS 向导迁移 Access 数据到 SQL Server，操作步骤如下：

(1) 在 SQL Server 企业管理器中的 Tools(工具)菜单上选择 Data Transformation。

(2) 选择 czdImport Data(导入数据)。

(3) 在 Choose a Data Source(选择数据源)对话框中选择 Microsoft Access as the Source，然后键入你的.mdb 数据库(.mdb 文件扩展名)的文件名或通过浏览寻找该文件。

(4) 在 Choose a Destination(选择目标)对话框中选择 Microsoft OLE DB Provider for SQL Server，选择数据库服务器，然后单击必要的验证方式。

(5) 在 Specify Table Copy(指定表格复制)或 Query(查询)对话框中单击 Copy Tables(复制表格)。

(6) 在 Select Source Tables(选择源表格)对话框中单击 Select All(全部选定)，然后单击“下一步”按钮，完成。

2. 使用 Transact-SQL 语句进行导入/导出

1) 在 SQL Server 里查询 Access 数据

```
SELECT * FROM OpenDataSource( 'Microsoft.Jet.OLEDB.4.0',
```

```
'Data Source="c:\DB.mdb";User ID=Admin;Password=')...表名
```

2）将 Access 导入 SQL Server

在 SQL Server 里运行：

```
SELECT * INTO newtable FROM OPENDATASOURCE ('Microsoft.Jet.OLEDB.4.0',
'Data Source="c:\DB.mdb";User ID=Admin;Password=')...表名
```

3）将 SQL Server 表里的数据插入到 Access 表中

在 SQL Server 里运行：

```
insert into OpenDataSource( 'Microsoft.Jet.OLEDB.4.0',
'Data Source=" c:\DB.mdb";User ID=Admin;Password=')...表名
(列名 1,列名 2)
select 列名 1,列名 2 from sql 表
```

例如：

```
insert into OPENROWSET('Microsoft.Jet.OLEDB.4.0','C:\db.mdb';'admin';'', Test)
select id,name from Test
INSERT INTO OPENROWSET('Microsoft.Jet.OLEDB.4.0', 'c:\trade.mdb'; 'admin'; ''表名)
SELECT * FROM sqltablename
```

3. SQL Server 和 Excel 的数据导入/导出

1）在 SQL Server 里查询 Excel 数据

```
SELECT * FROM OpenDataSource( 'Microsoft.Jet.OLEDB.4.0',
'Data Source="c:\book1.xls";User ID=Admin;Password=;Extended properties=Excel 5.0')...
[Sheet1$]
```

下面是一个查询的示例，它通过用于 Jet 的 OLE DB 提供程序查询 Excel 电子表格。

```
SELECT * FROM OpenDataSource ( 'Microsoft.Jet.OLEDB.4.0',
'Data Source="c:\Finance\account.xls";User ID=Admin;Password=;Extended properties=Excel
5.0')...xactions
```

2）将 Excel 的数据导入 SQL Server

```
SELECT * into newtable FROM OpenDataSource( 'Microsoft.Jet.OLEDB.4.0',
'Data Source="c:\book1.xls";User ID=Admin;Password=;Extended properties=Excel 5.0')...
[Sheet1$]
```

例如：

```
SELECT * into newtable FROM OpenDataSource( 'Microsoft.Jet.OLEDB.4.0',
'Data Source="c:\Finance\account.xls";User ID=Admin;Password=;Extended properties=Excel 5.0'
)...xactions
```

3）将 SQL Server 中查询到的数据导出成一个 Excel 文件

T-SQL 代码：

```
EXEC master..xp_cmdshell 'bcp 库名.dbo.表名 out c:\Temp.xls -c -q -S"servername" -U"sa" -P""'
```

参数：S 是 SQL 服务器名；U 是用户；P 是密码。

说明：还可以导出文本文件等多种格式。

例如：

```
EXEC master..xp_cmdshell 'bcp saletesttmp.dbo.CusAccount out c:\temp1.xls -c -q -S"
pmserver" -U"sa" -P"sa"'
EXEC master..xp_cmdshell 'bcp "SELECT au_fname, au_lname FROM pubs..authors ORDER BY au_lname"
queryout C:\ authors.xls -c -Sservername -Usa -Ppassword'
```

4）在 VB 6 中应用 ADO 导出 Excel 文件代码

```
Dim cn As New ADODB.Connection
cn.open "Driver = {SQL Server};Server = WEBSVR; DataBase = WebMis;UID = sa; WD = 123;"
cn.execute "master..xp_cmdshell 'bcp "SELECT col1, col2 FROM 库名.dbo.表名" queryout E:\DT.
xls -c -Sservername -Usa -Ppassword'"
```

5）在 SQL Server 里向 Excel 插入数据

```
insert into OpenDataSource( 'Microsoft.Jet.OLEDB.4.0',
'Data Source = "c:\Temp.xls";User ID = Admin;Password = ;Extended properties = Excel 5.0')...
table1 (A1,A2,A3) values (1,2,3)
```

T-SQL 代码：

```
INSERT INTO OPENDATASOURCE('Microsoft.JET.OLEDB.4.0',
'Extended Properties = Excel 8.0;Data source = C:\training\inventur.xls')...[Filiale1 $ ]
(bestand, produkt) VALUES (20, 'Test')
```

注：如果在 SQL Server 中查询 Excel 文件时出现问题：

```
SELECT *  FROM OPENROWSET( 'MICROSOFT.JET.OLEDB.4.0','Excel 8.0;IMEX = 1;HDR = YES;DATABASE = D:\
a.xls',[sheet1 $ ])
```

SQL Server 阻止了对组件 Ad Hoc Distributed Queries 的 STATEMENT 'OpenRowset/OpenDatasource 的访问，因为此组件已作为此服务器安全配置的一部分而被关闭。系统管理员可以通过使用 sp_configure 启用 Ad Hoc Distributed Queries。有关启用 Ad Hoc Distributed Queries 的详细信息，请参阅 SQL Server 联机丛书中的"外围应用配置器"。

可以通过启用 Ad Hoc Distributed Queries 来解决此问题。

启用 Ad Hoc Distributed Queries：

```
EXEC sp_configure 'show advanced options',1
reconfigure
EXEC sp_configure 'Ad Hoc Distributed Queries',1
reconfigure
```

使用完成后，关闭 Ad Hoc Distributed Queries：

```
EXEC sp_configure 'Ad Hoc Distributed Queries',0
reconfigure
EXEC sp_configure 'show advanced options',0
reconfigure
```

9.6 本章小结

本章主要介绍了 Access 和 SQL Server 数据库的常用使用方法。要求熟练掌握 Access 数据库表的设计制作，并对表之间的关系有清楚的认识。要求能够对数据库进行查询。熟练安装 SQL Server 服务器，能够启动、暂停和停止 SQL Server 服务器，并且解决 SQL Server 挂起。

要求对 SQL 语句熟练掌握，如创建、查询、插入、修改、删除数据，并且能够将语句运用到具体的项目中。能够进行 Access、SQL Server、Excel 数据之间的导入与导出，并且能够使用界面备份与恢复 Access 和 SQL Server 数据库。

9.7 思考与练习

1. 简述 Access 数据库的安装过程。
2. 利用 Access 创建 student(id,name,sex,tel),course(courseid,coursename)。
3. 使用 SQL Server 创建在线考试系统的数据库、表、记录。
4. 将 Access 创建好的数据库导出到 Excel 中，将 Excel 数据导入到 Access 中。

第10章 ASP案例——在线考试系统设计

在线考试系统使学生通过校园网或互联网访问网站，可以按照教师的建议或各自的计划对所学知识进行自我测验。教师可通过网站提供的环境对学生的学习情况进行动态管理，并依据网站记录的考试成绩给出对每一位学生的客观的综合评价，同时可进行出题命题工作，减轻教师的工作负担及提高工作效率。在线考试系统提高了考试的质量，从而使考试更趋于公证、客观，更加激发学生的学习兴趣。

学习目标及要求

(1) 分析在线考试系统的需求。

(2) 能够进行在线考试系统的模块设计。

(3) 能够实现数据库设计。

(4) 利用工具软件，实现前台和后台数据库的连接。

(5) 实现每一个功能模块，并自行测试。

10.1 系统设计

传统的考试方式一般要经过人工出卷、考生考试、人工阅卷等过程。对于一些课程来说，随着考生数量的增加，教师出卷、阅卷的工作量将会越来越大，其工作容易出错。在当今教育信息化的趋势及我国高校教育信息化系统的建设的大背景下，在线考试系统应运而生，其目的是充分利用学校现有的计算机软、硬件和网络资源实现无纸化考试，以弥补传统考试的不足。与传统考试模式相比，在线考试渗入了更多的技术环节，对实现安全性的途径、方法也提出了更高的技术要求。通过 Internet 来实现在线考试，是现代教育技术的一种体现，具有很重要的现实意义。在线考试可以实现教考分离以及考务工作的全自动化管理；可以有效利用校园网的软硬件资源，使其发挥最大效应，更好地为学校的教学、科研、管理服务；可以大规模地实行考试，实现考试的客观性、公正性；通过自动化组卷、阅卷，可以减轻教师的工作强度；可以利用网络的无限广阔空间，随时随地对学生进行考试；此外，由于增加了 Web 数据库技术的应用，可以简化考试的流程。

10.1.1 需求分析

要完成一个软件系统，首先应该进行需求分析。在搞清楚用户的需要及对系统使用的功能要求的情况下，才能设计出令用户满意的产品。依据需求，在线考试系统应具有以下

功能：

(1) 用户信息的管理。管理员可以增加、删除、修改信息。

(2) 试卷设置。设置试题每种题型的数量和分值以及考试的总时间。

(3) 试题库管理。创建试题库，通过计算机自动组卷。试卷内容应该包括选择、判断、填空、简答 4 种题型，实现对试题库的管理，使试题的增删、编辑更为简便。

(4) 在线考试。系统控制整个考试过程，实现时间的监控与权限的控制，考生在限定的考试时间内答卷。

(5) 计算机自动阅卷。本系统只考虑客观题，要求计算机能自动阅卷，并能及时显示出考生分数。

(6) 成绩查阅。考生考完以后，管理员对所有记录进行查询，同时可以删除指定记录。

10.1.2 开发运行环境

本章介绍的在线考试系统采用 ASP(active server pages)语言进行开发。ASP 是微软开发的一种类似于 HTML、Script(脚本)与 CGI 的结合体，ASP 指令与 ActiveX 配合建立动态、交互且高效的 Web 服务器应用程序。有了 ASP 就不必担心客户的浏览器是否能运行你所编写的代码，因为所有的程序都将在服务器端执行，包括所有嵌在普通 HTML 中的脚本程序。当程序执行完毕后，服务器将执行的结果以 HTML 格式传送给客户浏览器，这样，不仅仅提高了程序的灵活性，也减轻了客户端浏览器的负担，大大提高了交互的速度。

本程序的开发与运行环境如下。

- 操作系统：Windows XP Professional。
- 数据库管理系统：Microsoft SQL Server。
- 开发工具：Dreamweaver。
- 服务运行：IIS 3.0 以上。

10.1.3 功能模块设计

为了增强管理效果和信息的及时更新，按照以下功能需求对系统进行分析。

1. 用户管理功能

用户分为管理员、教师用户、学生用户 3 类。不论是管理员、教师或学生，都需要通过用户名和口令进行登录。管理员和教师用户可以对所有的学生成绩进行增加、删除和修改。学生用户只能看到自己的学习成绩，不能修改。由于管理员、教师用户及学生用户较为固定，教师和学生信息的管理可以由系统管理人员直接操作数据库完成。

2. 教师对成绩的管理功能

教师对成绩的管理可以从两个角度进行：按照课程角度进行管理和按照学生角度进行管理。教师登录该系统以后，可以对这两种方式进行选择。

1) 按照课程角度进行管理

按照课程角度管理，教师可以看到课程信息和成绩信息。课程信息包括课程编号、课程

名称、学时和课程类型。当教师双击某一课程时，可以自动显示出该课程的所有学生的成绩信息，成绩信息包括选修这门课的学生的学号、姓名、班级和考试成绩；并同时自动计算出所有学生的人数，以及所有学生的平均成绩。在成绩信息中，教师可以对学习成绩进行增加、修改和删除，并能够打印报表。

2）按照学生角度进行管理

按照学生角度管理，教师可以看到学生信息和成绩信息。学生信息包括学号、姓名、班级、专业。当教师双击某一学生信息时，可以自动显示出该课程的这个学生所有课程的学习成绩，并同时自动计算出这个学生的总分和平均分。在成绩信息中，教师可以对学习成绩进行增加、修改和删除，并能够打印报表。

3. 学生对成绩的查看功能

学生用户登录后，可以看到自己的学生信息和成绩信息，并能够打印报表。学生信息包括学号、姓名。成绩信息包括这个学生的所有课程的成绩信息、总分及平均分。

学生成绩管理模块功能图如图 10-1 所示。

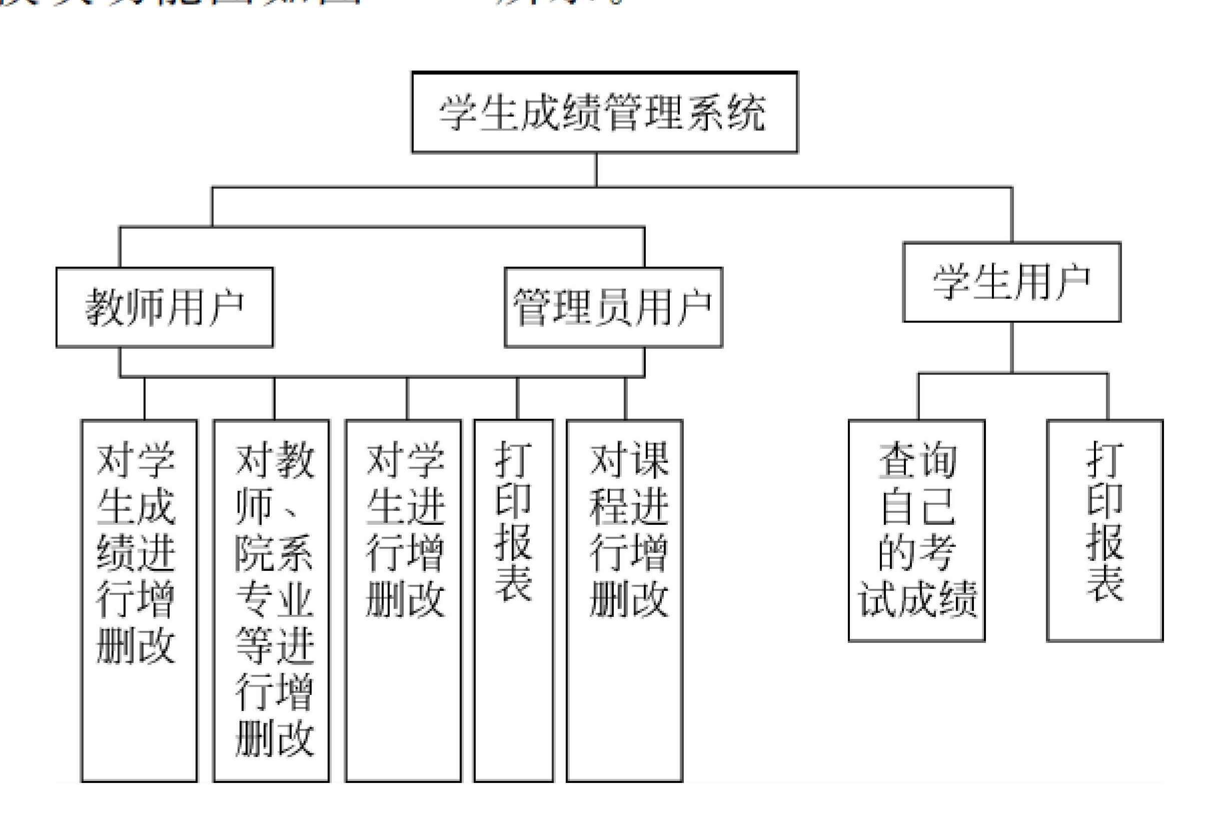


图 10-1 学生成绩管理模块功能图

10.1.4 数据库设计

针对一般学生成绩管理信息系统的需求，通过对成绩管理过程的内容和数据流程分析，设计出该系统的数据项和数据结构。

1. 系统数据项及结构

(1)“学生信息”的数据项有学生学号、学生姓名、班级名称、登录密码等。

(2)“课程信息”的数据项有课程代号、课程名称、学时数等。

(3)“学生成绩信息”的数据项有学生学号、考试时间、课程代号、成绩等。

(4)“选课信息”的数据项有专业代号、课程代号。

(5)“班级信息”的数据项有班级代号、班级名称、专业代号。

(6)“教师信息”的数据项有教师工号、教师姓名、登录密码。

(7)“管理员信息”的数据项有管理员姓名、登录密码。

(8) “专业信息”的数据项有专业代号、专业名称。

(9) “授课信息”的数据项有教师工号、课程代号、班级代号。

(10) “题库信息”的数据项有试卷代号、试卷名称、教师工号。

(11) “选题表”的数据项有班级代号、课程代号、试卷代号、考试时间、开始时间、结束时间。

(12) “未阅试题表”的数据项有学生学号、课程代号、题号、答案、成绩。

(13) “已阅试题表”的数据项有学生学号、课程代号、试题类型、答案。

根据上面的设计可以规划出的实体有学生实体、课程实体和学生成绩实体、班级实体、院系实体、教师实体、管理员实体、专业实体。

2. 系统 E-R 图(如图 10-2 所示)

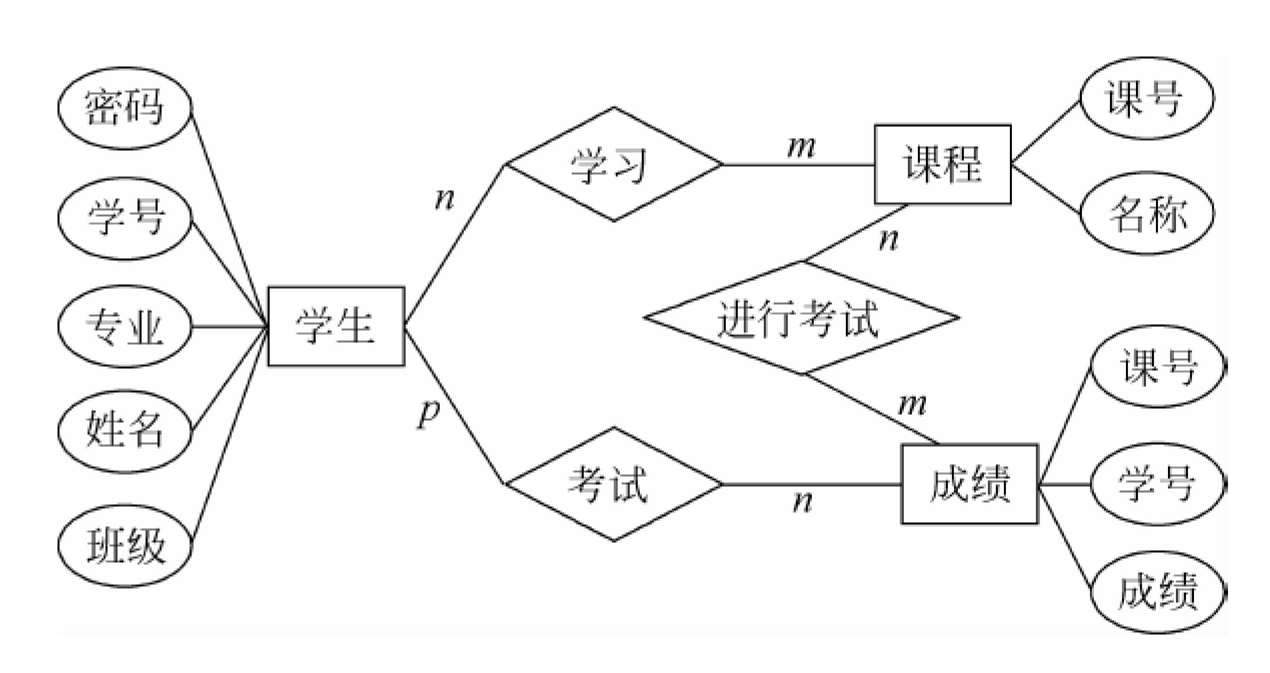


图 10-2 系统 E-R 图

3. 系统逻辑设计

本系统是在 SQL Server 2000 数据库管理系统平台创建的，主要由 13 张表组成：SuperInfo、MajorInfo、CouInfo、ClassInfo、CouChoose、TeachCou、TeachInfo、StuInfo、PaperList、StuScore、PaperChoose、PaperUnCheck、PaperChecked。

(1) SuperInfo 表，即管理员信息表，如表 10-1 所示。

表 10-1 SuperInfo

字段名称	内容	类型	长度	备注
SuperName	管理员姓名	Varchar	50	主键
SuperPwd	登录密码	Varchar	50	

(2) MajorInfo 表，即专业信息表，如表 10-2 所示。

表 10-2 MajorInfo

字段名称	内容	类型	长度	备注
MajorNum	专业代号	Varchar	20	主关键字
MajorName	专业名称	Varchar	50	

MajorNum(专业号)字段为该表的主关键字，唯一地标识了一个专业的记录，与学生信息表、选课表中的 MajorNum 字段相关联。

（3）CouInfo 表，即课程信息表，如表 10-3 所示。

表 10-3　CouInfo

字段名称	内　　容	类　　型	长　　度	备　　注
CouNum	课程代号	Varchar	20	主关键字
CouName	课程名称	Varchar	50	
CouBook	学时数	Varchar	50	

CouNum 作为此表的主键，唯一标识一门课程，与选课表、授课表、选题表、未阅试题记录表、已阅试题表、成绩表中的 CouNum 关联。

（4）ClassInfo 表，即班级信息表，如表 10-4 所示。

表 10-4　ClassInfo

字段名称	内　　容	类　　型	长　　度	备　　注
ClassNum	班级代号	Varchar	12	主键
ClassName	班级名称	Varchar	50	
MajorNum	专业代号	Varchar	50	外键

MajorNum 作为本表的外键与专业表中的 MajorNum 关联（级联删除和更新）。

（5）CouChoose 表，即选课表，如表 10-5 所示。

表 10-5　CouChoose

字段名称	内　　容	类　　型	长　　度	备　　注
MajorNum	专业代号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键

（6）TeachCou 表，即授课表，如表 10-6 所示。

表 10-6　TeachCou

字段名称	内　　容	类　　型	长　　度	备　　注
TeachNum	教师工号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键
ClassNum	班级代号	Varchar	20	外键

（7）TeachInfo 表，即教师信息表，如表 10-7 所示。

表 10-7　TeachInfo

字段名称	内　　容	类　　型	长　　度	备　　注
TeachNum	教师工号	Varchar	20	主关键字
TeachName	教师姓名	Varchar	50	
TeachPWD	登录密码	Varchar	30	

TeachNum 作为此表的主关键字与授课表和题库表的 TeachNum 字段外键关联。

(8) StuInfo 表,即学生信息表,如表 10-8 所示。

表 10-8 StuInfo

字段名称	内容	类型	长度	备注
StuNum	学生学号	Varchar	20	主关键字
StuName	学生姓名	Varchar	20	
MajorNum	专业代号	Varchar	20	外键
ClassNum	班级代号	Varchar	20	外键
StuPwd	登录密码	Varchar	30	
LoginOrNot	登录状态	Varchar	10	

(9) PaperList 表,即题库表,如表 10-9 所示。

表 10-9 PaperList

字段名称	内容	类型	长度	备注
PaperNum	试卷代号	Varchar	100	主关键字
PaperName	试卷名称	Varchar	100	
TeachNum	教师工号	Varchar	20	

(10) StuScore 表,即成绩表,如表 10-10 所示。

表 10-10 StuScore

字段名称	内容	类型	长度	备注
StuNum	学生学号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键
TestDate	考试时间	Varchar	11	
Score	成绩	Varchar	3	

(11) PaperChoose 表,即选题表,如表 10-11 所示。

表 10-11 PaperChoose

字段名称	内容	类型	长度	备注
ClassNum	班级代号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键
PaperNum	试卷代号	Varchar	100	
TestDate	考试时间	Varchar	20	
StartTime	开考时间	Varchar	20	
EndTime	结束时间	Varchar	20	

（12）PaperUnCheck 表，即未阅试题记录表，如表 10-12 所示。

表 10-12　PaperUnCheck

字段名称	内　容	类　型	长　度	备　注
StuNum	学生学号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键
TermNum	题号	Varchar	10	
StuAnswer	答案	Varchar	1000	
Score	成绩	Varchar	3	

（13）PaperChecked 表，即已阅试题表，如表 10-13 所示。

表 10-13　PaperChecked

字段名称	内　容	类　型	长　度	备　注
StuNum	学生学号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键
TermType	试题类型	Varchar	10	
StuAnswer	答案	Varchar	1000	

10.2　系统实现

下面给出系统的详细界面，以表述在线考试管理系统的具体功能。由于设置了 3 种用户的不同权限，所以有 3 种不同的身份登录地址。

10.2.1　管理员权限

登录地址：http://localhost/zxks/back/adminLogin.asp。

首先是进入管理员登录界面，如图 10-3 所示，用户名输入“admin”，密码初设为“admin”，验证码输入“8842”。本页面显示系统名称、管理员用户名录入、密码校验、验证码信息。尤其是验证码，采用自动随机生成技术。当管理员输入姓名、密码、对应的验证码信息之后，单击“提交”按钮就会将输入信息提交给系统后台数据库，以验证用户身份。如果用户身份正确，则会进入系统的管理员控制界面。



图 10-3　管理员登录界面

在登录过程中，不管是一般用户还是管理员，都使用各自设定的用户名和密码来登录系统。如果在登录过程中，所登录的用户名和密码中任何一个与原始数据不一致，将不能正常地登录系统，直到所输入的数据与原始数据完全一致为止。具体实现代码略。

管理员级别用户登录进入系统后拥有对该系统的专业、班级、教师、学生、课程等信息进行增加、删除和修改的权限。

1. 专业操作

在考试系统中，管理员初次登录，第一件必须要做的事情就是添加全校的专业信息，有了专业信息，接下来就可设定班级及学生信息。所以在本系统中，初次登录者首先要添加专业信息，如图 10-4 所示。需要设置出专业代码和名称，可根据学校的安排设置。同时，在添加完多个专业之后，系统还提供了对所有已经设置的专业进行修改和删除的操作。一旦教务人员输入出错，就可以对相应的专业名称进行修改，具有很强的可操作性。

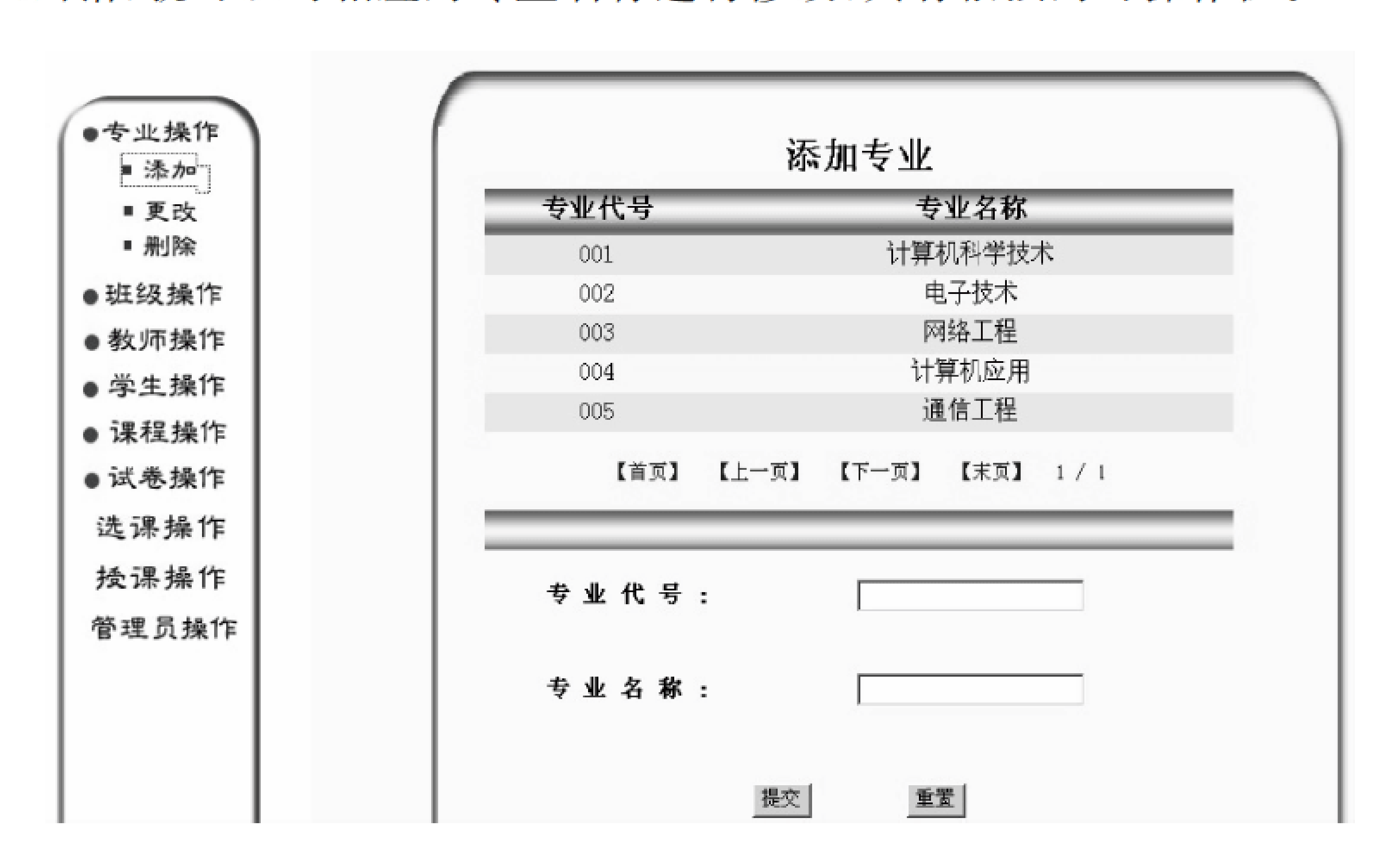


图 10-4　添加专业界面

2. 班级操作

在系统中，专业操作完毕之后，才能在已有的专业基础上添加班级，如图 10-4 所示。可从计算机科学技术、电子技术、网络工程、计算机应用、通信工程中选择一个专业，然后在此专业基础上添加班级的代号和名称。在班级操作中可进行添加、删除、修改。本系统提示了“注意：请选择专业”的消息，如图 10-5 所示。

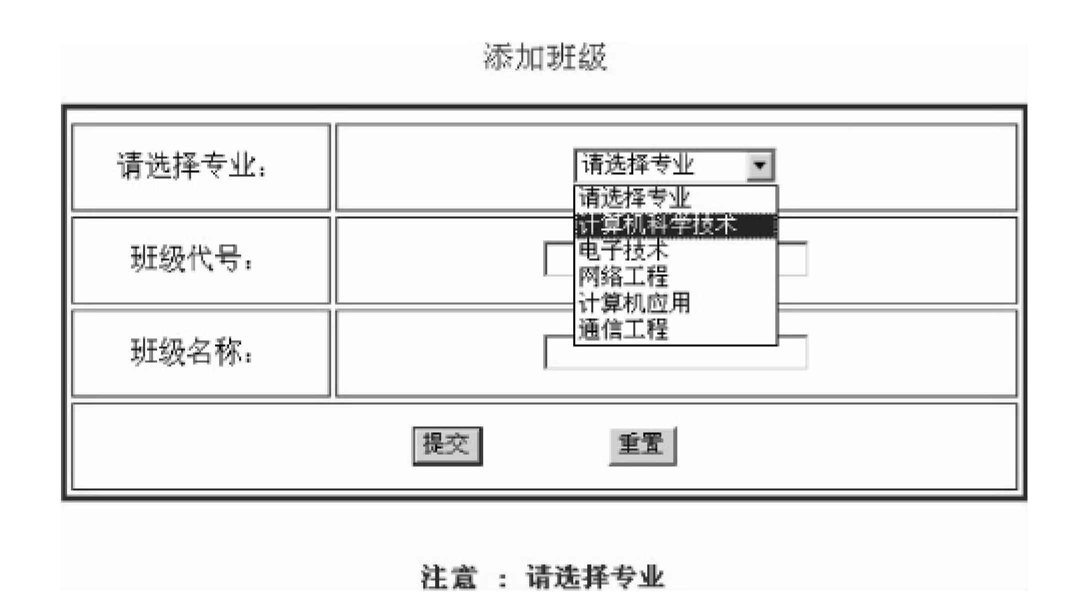


图 10-5　添加班级界面

3. 教师操作

在系统中，可以对教师信息进行添加、修改、删除操作。如图 10-6 所示，设置教师工号、姓名、密码。

添加教师信息

教师工号:	
教师姓名:	
教师密码:	
提交　重置	

图 10-6　添加教师信息界面

如图 10-7 所示,教师信息删除界面列出了某大学部分教师的名单,可根据教师实际情况增添、删除相应的教师信息,给教师管理提供了方便。

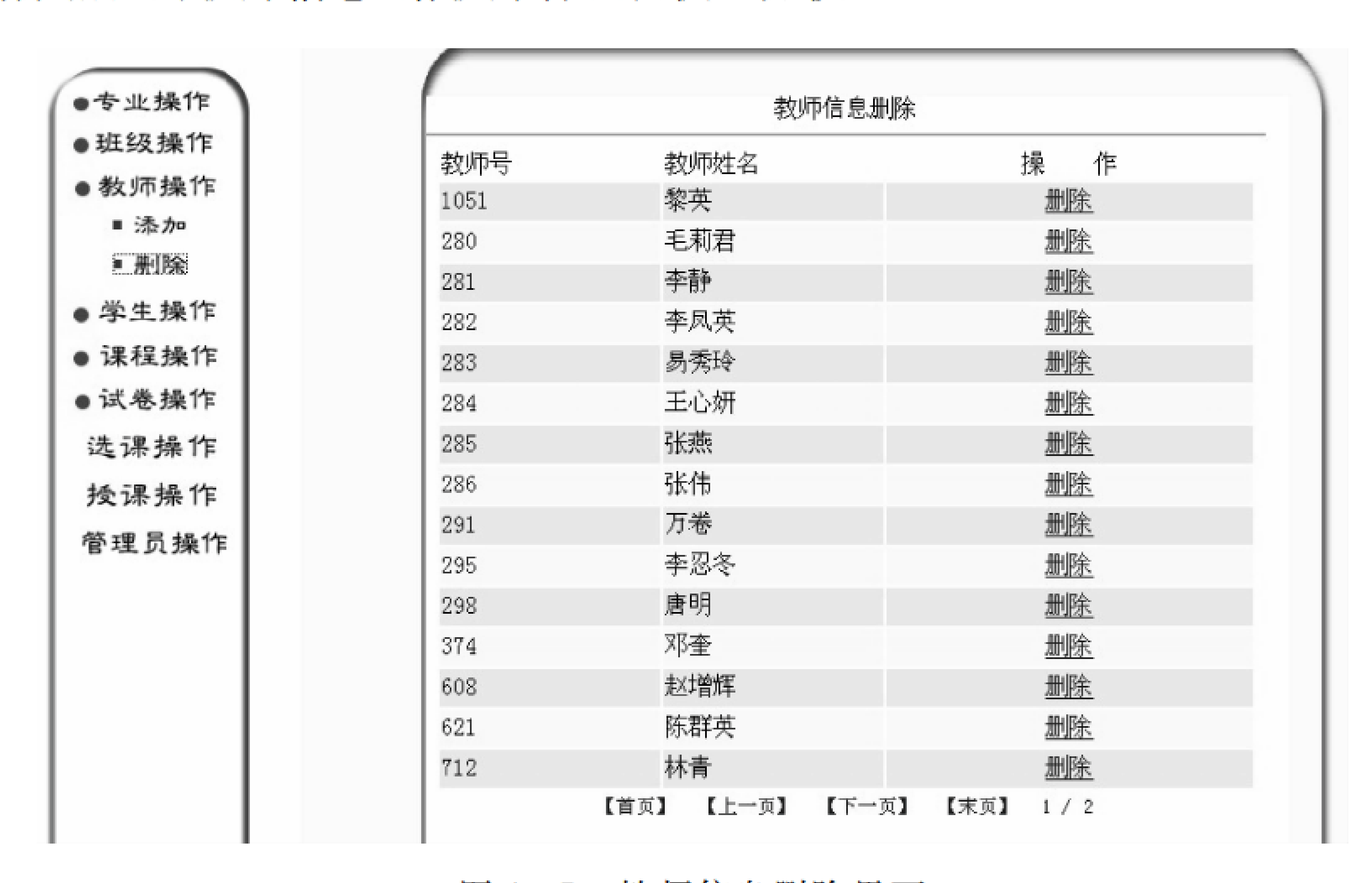


图 10-7　教师信息删除界面

4. 课程操作

在课程操作中,可以添加各种不同院系的课程。要求是:首先要给教师分配授课任务,每个教师会有固定的几门课程,其次才可以给教师选择课程,最后该教师登录系统后,可以为自己所授课程出题。添加课程界面如图 10-8 所示,其中列出了课程代号、课程名称、课程用书等信息,同时提示“请填写完整信息”的消息。

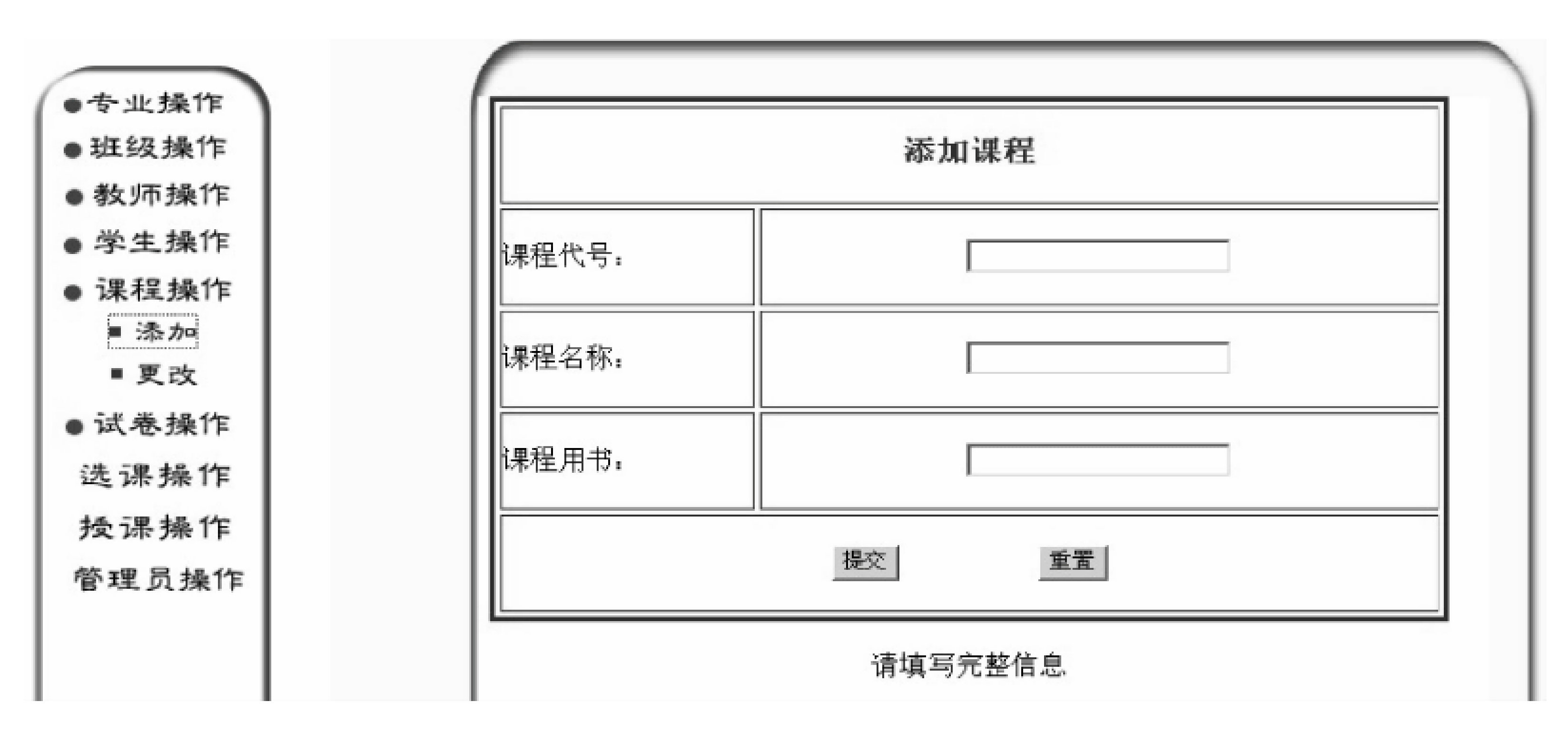


图 10-8　添加课程界面

通过课程信息添加与删除界面可以查看已经添加的课程，也可以修改课程名称和所选教材，还可以对已有的课程进行删除，如图 10-9 所示。该界面提供上翻和下翻功能。

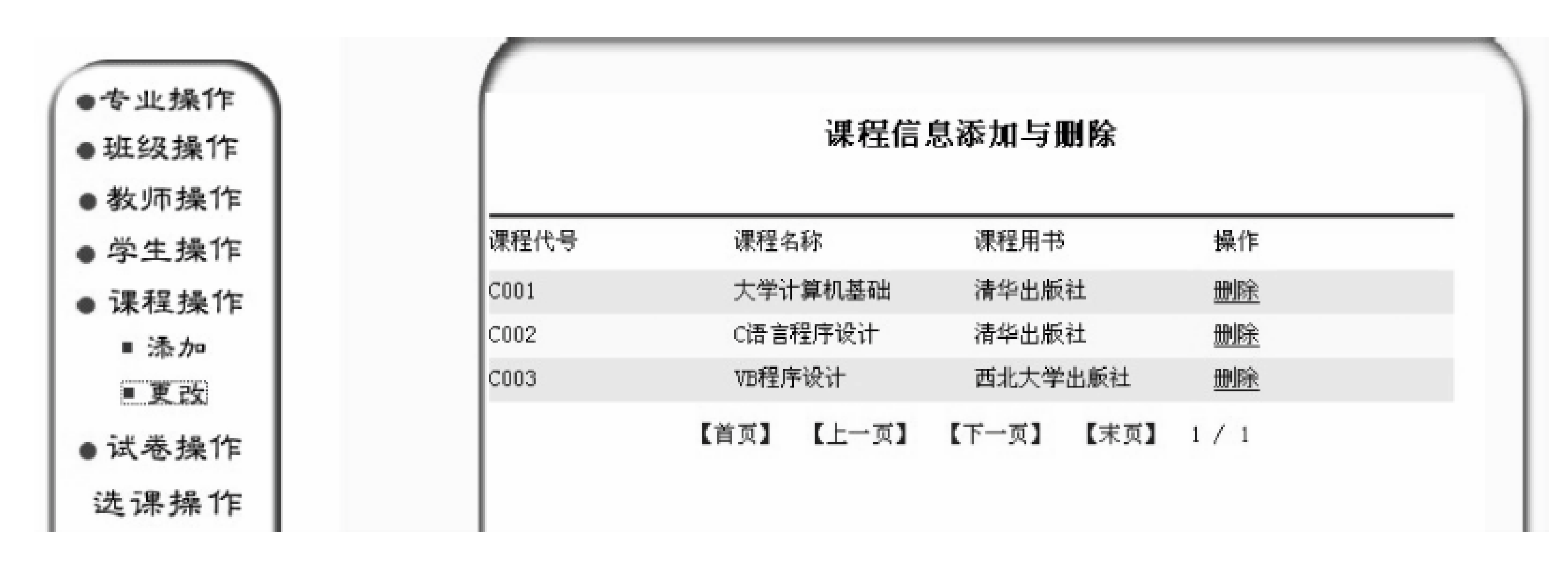


图 10-9 课程信息添加与删除界面

通过授课操作界面可以选择课程，并分配给已经添加的教师，还可以删除授课记录，如图 10-10 所示。该界面将删除与添加操作合成在一个界面中。该界面提供上翻、下翻功能。

授课操作

班级	课程	教师	操作
7z0921	大学计算机基础	赵增辉	删除
7z0921	C语言程序设计	唐明	删除
7z0921	VB程序设计	张伟	删除
7z0931	大学计算机基础	黄玉雷	删除
7z0931	C语言程序设计	李忍冬	删除
7z0931	VB程序设计	万卷	删除
7z0941	大学计算机基础	李艳	删除
7z0941	C语言程序设计	林青	删除
7z0941	VB程序设计	赵增辉	删除

【首页】 【上一页】 【下一页】 【末页】 1 / 1

班级： 请选择班级

课程：

教师： 黎英

提交 重置

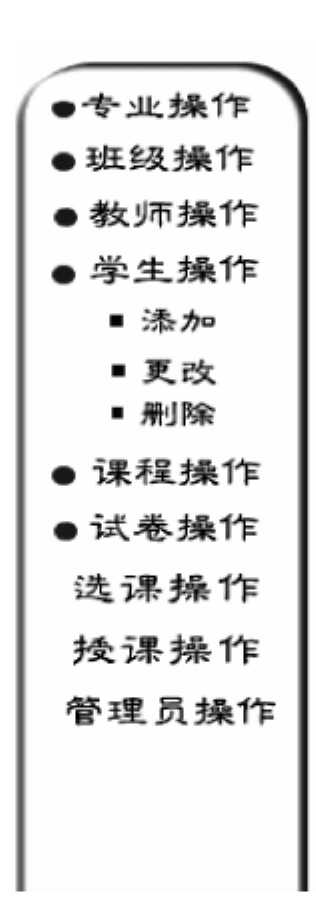
图 10-10 授课操作界面

在操作时注意，必须先添加教师，之后添加课程，然后才能给教师分配相应的授课任务，否则在后面出题过程中，该教师无法出题。

5. 学生操作

学生操作之前，要选择相应的专业和班级。输入的信息有学生学号、姓名、所选专业和班级以及登录密码。添加学生信息界面如图 10-11 所示。在专业设置中，如果没有设置对应的专业，那么班级信息是无法给出的。例如，设置计算机科学与技术专业，选择对应班级 4Z0711 班。如果信息出错，会出现“请将信息填写完整”的提示信息。

通过更改学生信息和删除学生信息查看已经添加的学生名单。系统的更改功能可以修改学生姓名、专业、班级以及登录密码等信息。系统添加锁定功能，如果限制某位考生参加



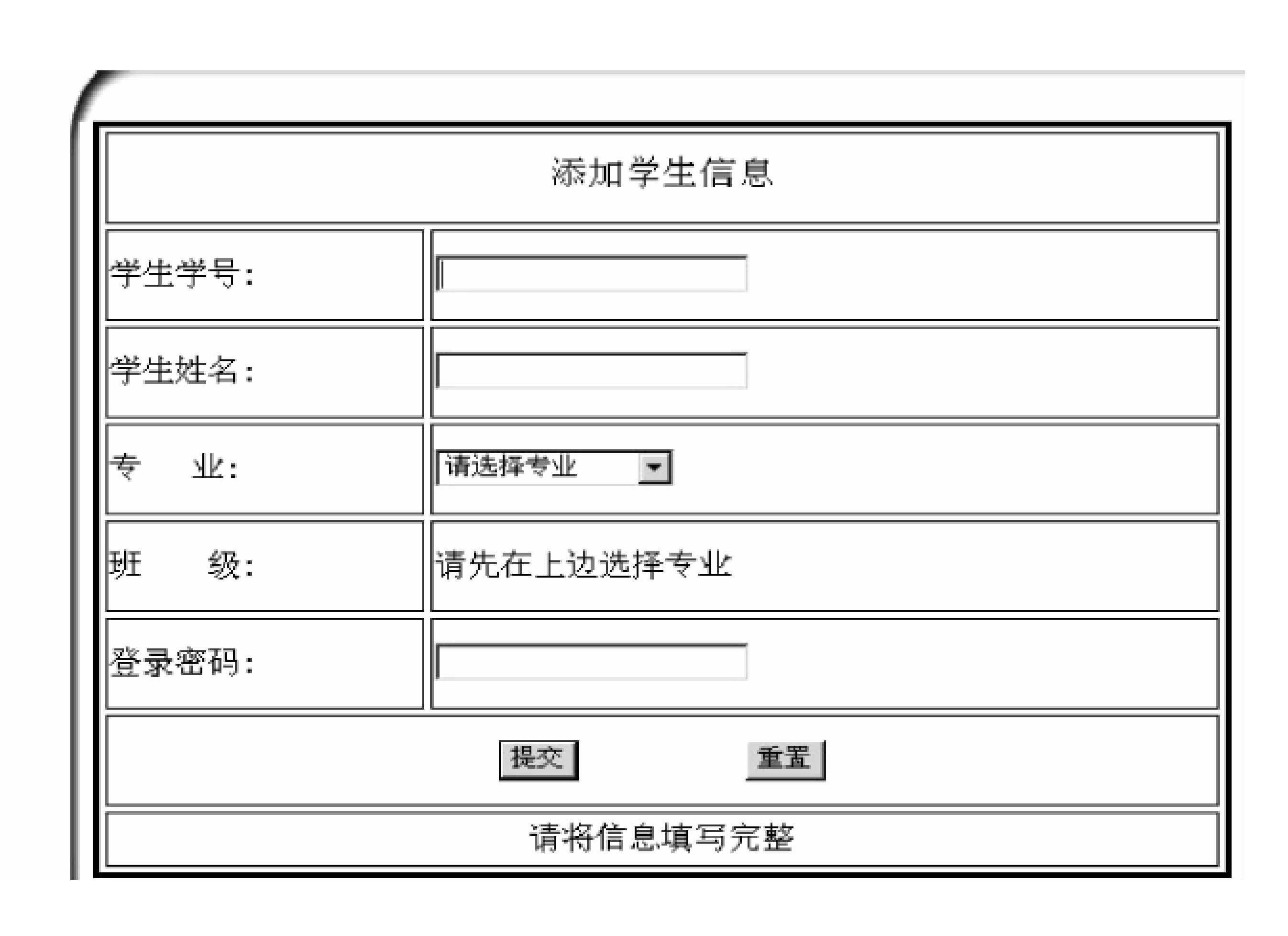


图 10-11　添加学生信息

考试，可将其“解锁”状态选择为“锁定”状态；如果允许某位考生参加考试，可将其“锁定”状态切换到“解锁”状态。更新学生信息界面如图 10-12 所示。



学生学号　学生姓名	所选专业	所在班级	登入密码	解锁
7z092101马梦轩	计算机科学技术	7z0921	7z092101	解锁
7z092103丁海东	计算机科学技术	7z0921	7z092103	锁定
7z092104梁小龙	计算机科学技术	7z0921	7z092104	锁定
7z093101杜超	计算机科学技术	7z0931	7z093101	锁定
7z093102葛忠平	计算机科学技术	7z0921	7z093102	锁定
7z093103丛学志	计算机科学技术	7z0931	7z093103	锁定
7z094101王普琛	计算机科学技术	7z0941	7z094101	锁定
7z094102刘图	计算机科学技术	7z0941	7z094102	锁定
7z094103侯鑫	计算机科学技术	7z0941	7z094103	锁定

图 10-12　更新学生信息

10.2.2　教师权限

1. 教师登录

在管理员添加了教师信息后，教师用户就可以登录系统，为自己所授课程进行出题了，教师用户登录地址为 http://localhost/zxks/back/teacherLogin.asp。

教师登录界面如图 10-13 所示，教师可以输入自己的工号，并设置登录密码，用户名和密码将提交到数据库中进行校验。如教师工号：608，密码：608。



图 10-13　教师登录界面

2. 试卷创建操作

教师输入自己的工号和密码后就进入了试卷操作页面，如图 10-14 所示。

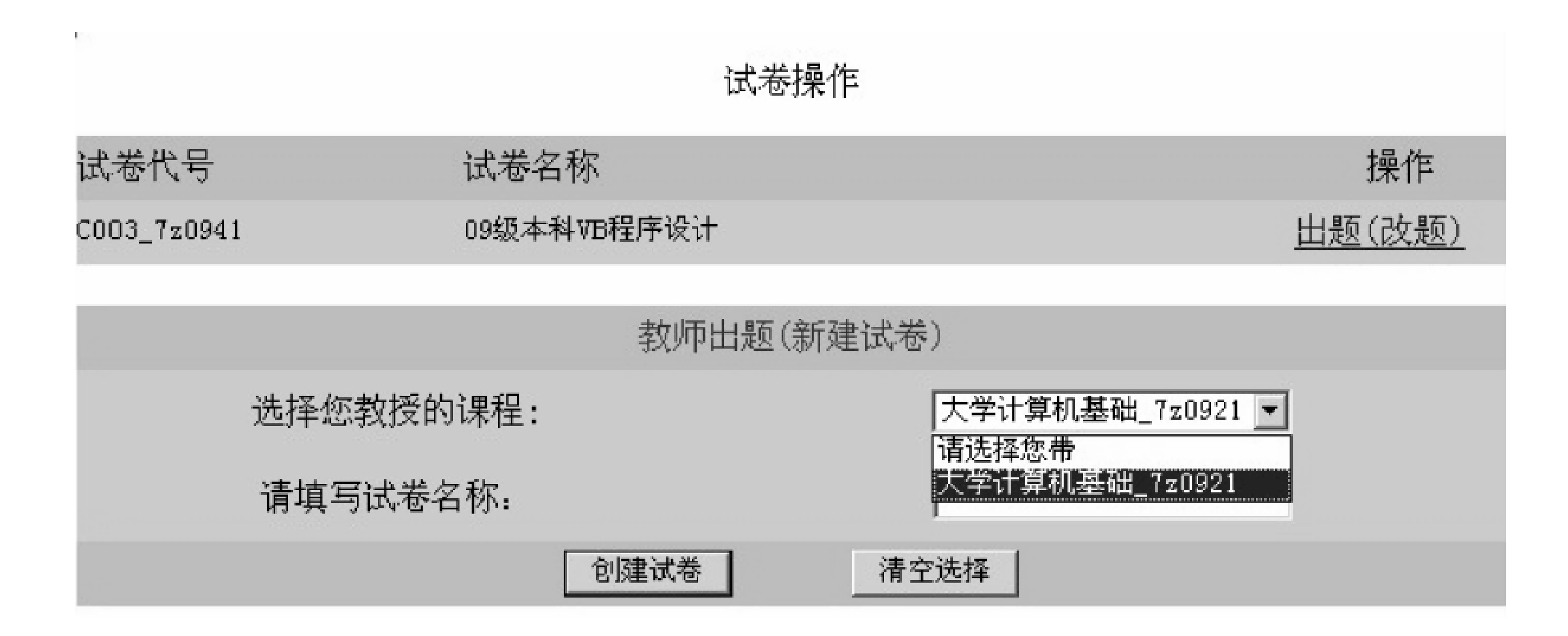


图 10-14 试卷操作界面

试卷代号、试卷名称数据类型可按照学校的标准要求设置。教师出题只能选择自己所授课程,不允许给其他课程出题。教师必须按照学校规范标准填写试卷名称,然后生成试卷。

3. 试卷出题操作

教师出题界面如图 10-15 所示。教师输入自己所授的课程,进行创建试卷操作,然后选择相应题目类型,便可在"给出题目内容"框输入自己设置的题目。

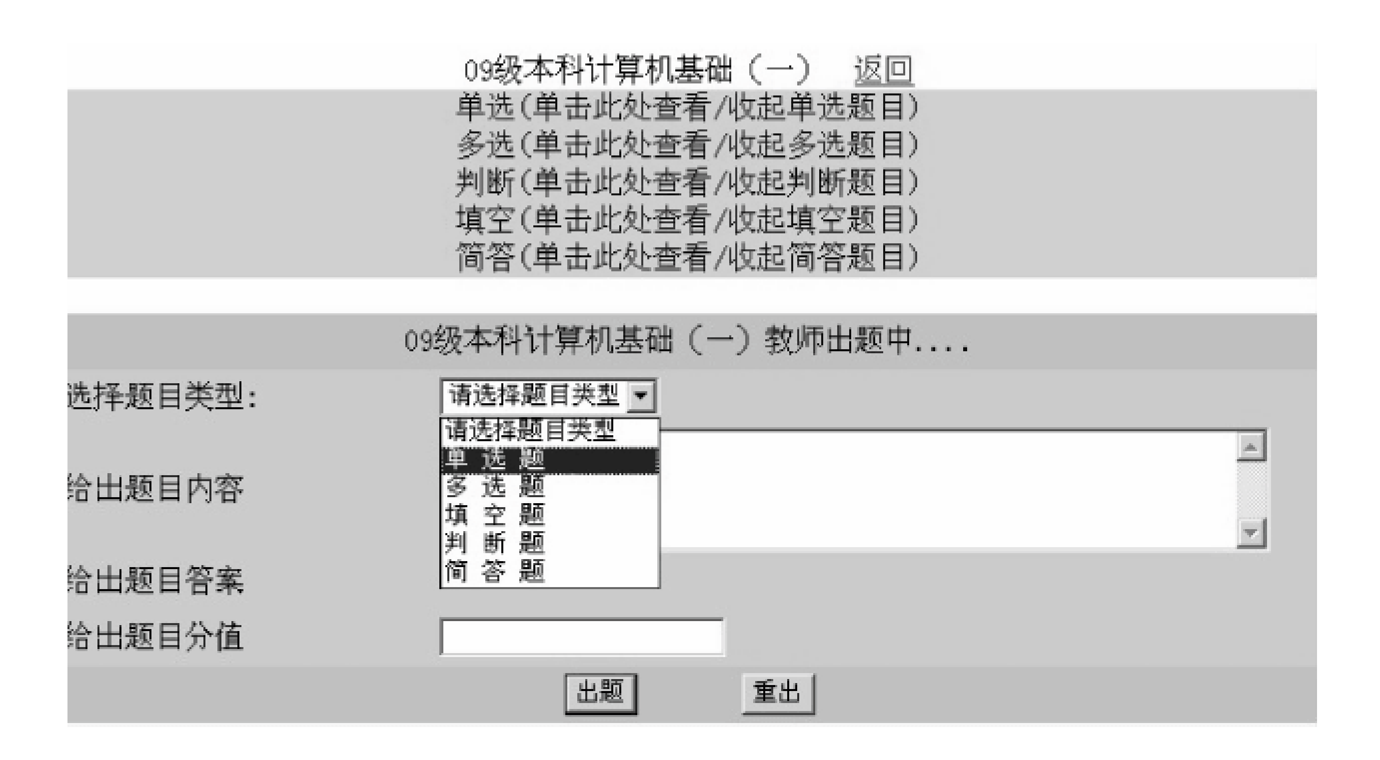


图 10-15 教师出题界面

在添加多种类型的考试题目后,可以单击"单击此处查看/收起××题目"进行查看,并可以修改题目。试题界面如图 10-16 所示。试题中有单选题、多选题、填空题、判断题、简答题。例如 2009—2010 年计算机基础试题,根据学院试题标准,设定了 A、B、C、D 选项格式,按照要求出题,并将答案给出。多选题里,除了格式和单选题一致以外,答案也必须给定,答案可设置多种形式。判断题里,直接给出题目的语句,然后在答案中给出 right 或 wrong。

教师在出题过程中,首先要选择题目的类型,确定好之后,再给出题目的内容,最后给出题目的正确答案。

如果出现错误的题目,教师可以自行修改,只须单击"单击此处查看"进行修改,或单击题目里的"删除"按钮,对题目进行删除。修改和删除后的题目会直接在数据库中进行更新操作。

2009-2010计算机基础试题　返回
单选(单击此处查看/收起多选题目)
S00　删除　word的后缀名是（）　A、TXT　B、MDB　C、xml　D、doc
答案：　A
S01　删除　在计算机中数据采用的是（　）进制。A. 十六进制　B. 十进制　C. 八进制　D. 二进制
答案：　D
多选(单击此处查看/收起多选题目)
M00　删除　计算机操作系统的主要作用是（）　A. 处理机管理　B. 设备管理　C. 作业管理　D. 进程管理
答案：　A, B, C, D
判断(单击此处查看/收起多选题目)
J00　删除　计算机系统是有软件系统和硬件系统构成的。　修改
答案：　right
填空(单击此处查看/收起填空题目)
简答(单击此处查看/收起简答题目)

2009-2010计算机基础试题教师出题中....
选择题目类型：　请选择题目类型
请选择题目类型
单 选 题
多 选 题
填 空 题
判 断 题
简 答 题
给出题目内容
给出题目答案

图 10-16　试题界面

单击“返回”按钮就回到了教师出题的界面，同时可以进行出题类别的选择以及科目的选择。

4. 试卷操作

出题操作成功后，管理员级别用户就可以给相应的班级选择相应的试题并设置考试时间等信息，为考试做好一切准备工作。考试时间设置界面如图 10-17 所示。

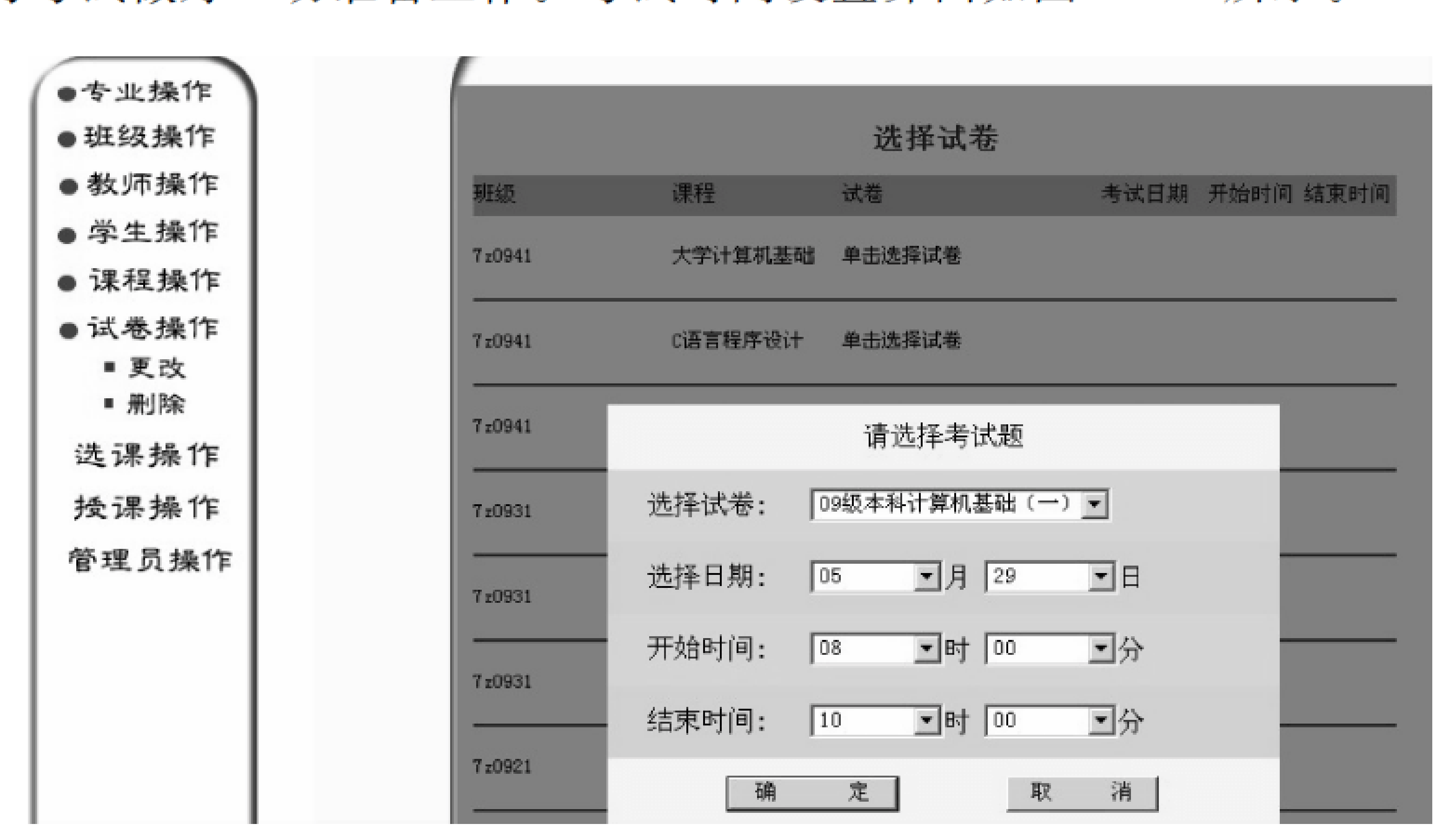


图 10-17　设置考试时间界面

教师出题成功后，教务人员就要开始对考试进行安排，即对不同专业、不同学生、不同课程进行考试时间的设定，这也是平时操作最复杂的。例如对于 4Z0941 班学生，只能在 5 月 29 日早晨 8：00～10：00 进行“09 本科计算机基础（一）”的考试，如果有学生超过这个时间答题，或没有赶上这个时间，那么他的考试成绩就为空，只有在这个时间段考试才是有效的。

在考试信息设置后，单击“确定”按钮，将弹出试卷修改成功提示框，如图 10-18 所示。



图 10-18　试卷修改成功确认界面

试卷设置成功后，出题教师所授课的班级就可以在设置好的时间进行考试。

10.2.3　学生权限

1. 学生登录

管理员添加学生后，该学生就可以作为考生以考生权限登录本系统。考生的登录地址为 http://localhost/zxks/index.asp。学生登录界面如图 10-19 所示。



图 10-19　学生登录界面

2. 学生答题

在学生登录界面，学生输入自己的学号和密码（初始密码就是学生的学号）后就进入了答题页面，如图 10-20 所示。



图 10-20　学生答题界面

学生完成答题后，可以登录学生成绩查询界面自行查询成绩。学生成绩查询界面如图 10-21 所示。教师登录成绩修改界面（如图 10-22 所示）后，通过班级、课程的输入，查找到该班级的学生成绩。在这里要强调的是所输入的数据必须与数据库中的数据类型保持一致，比如分数最低为 0 分，最高为 100 分，这个是数据库中给定的。

计算机应用技术2班 计算机应用基础

学号	姓名	课程	考试时间	得分
4z071201	于彬	计算机应用基础	2010-4-18	95

图 10-21 学生成绩查询界面

计算机应用技术2班成绩表

学号	姓名	课程	考试时间	得分	
4z071201	于彬	计算机应用基础	2010-4-18	95	修改 删除

图 10-22 教师成绩修改界面

10.3 系统运行

1. 在线考试系统后台数据库的建立代码

操作方法：打开 SQL Server 2000 数据库，在查询分析器中输入 SQL 语句，并运行。

以下 SQL 代码依次创建了 OnLinePaper 总数据库、管理员信息表、专业信息表、班级信息表、学生信息表、课程信息表、教师信息表、选课表等，这些表为在线考试系统提供了后台支持。

```
create database OnLinePaper
go

use OnLinePaper
go
--drop database OnLinePaper

--创建管理员信息表
create table SuperInfo
          (
          SuperName varchar(50) primary key,
          SuperPwd varchar(50) not null
          )
go
insert into SuperInfo values('louis','louis')
--创建专业信息表
create table MajorInfo
          (
          MajorNum varchar(20) primary key,
          MajorName varchar(50) not null
          )
go

--创建班级信息表
```

```
create table ClassInfo
        (
        ClassNum varchar(20) primary key,
        ClassName varchar(50) not null,
        MajorNum varchar(20)
        )
go

-- 创建学生信息表
create table StuInfo
        (
        StuNum varchar(20) primary key,
        StuName varchar(20) not null,
        MajorNum varchar(20) foreign key references MajorInfo(MajorNum) on delete cascade
on update cascade,
        ClassNum varchar(20) foreign key references ClassInfo(ClassNum) on delete cascade
on update cascade,
        StuPwd varchar(30) not null
        )
go

-- 创建课程信息表
create table CouInfo
        (
        CouNum varchar(20) primary key,
        CouName varchar(50) not null,
        CouBook varchar(50) not null
        )
go

-- 创建教师信息表
create table TeachInfo
        (
        TeachNum varchar(20) primary key,
            TeachName varchar(50) not null,
        TeachPWD varchar(30) not null
            )
go

-- 创建选课表
create table CouChoose
        (
        MajorNum varchar(20) foreign key references MajorInfo(MajorNum) on delete cascade
on update cascade,
         CouNum varchar(20) foreign key references CouInfo(CouNum) on delete cascade on
update cascade
        )
go

-- 创建授课表
create table TeachCou
```

```
            (
            TeachNum varchar(20) foreign key references TeachInfo(TeachNum) on delete cascade
on update cascade,
             CouNum varchar(20) foreign key references CouInfo(CouNum) on delete cascade on
update cascade,
            ClassNum varchar(20) foreign key references ClassInfo(ClassNum) on delete cascade
on update cascade
            )
go

-- 创建题库表
create table PaperList
            (
            PaperNum varchar(100) primary key,
            PaperName varchar(100) not null,
            TeachNum varchar(20) not null
            )
go

-- 创建选题表
create table PaperChoose
            (
            ClassNum varchar(20) foreign key references ClassInfo(ClassNum) on delete cascade
on update cascade,
             CouNum varchar(20) foreign key references CouInfo(CouNum) on delete cascade on
update cascade,
            PaperNum varchar(100) foreign key references PaperList(PaperNum) on delete cascade
on update cascade,
            TestDate varchar(20),
            StartTime varchar(20),
            EndTime varchar(20)
            )
go

-- 创建学生答题(答案)表
create table StuAnswer
            (
             StuNum varchar(20) foreign key references StuInfo(StuNum) on delete cascade on
update cascade,
             CouNum varchar(20) foreign key references CouInfo(CouNum) on delete cascade on
update cascade,
            AnswerFileName varchar(100) not null
            )
```

2. 前台和后台数据库连接代码(odbc_connection.asp)

创建 odbc_connection.asp 文件,在文件代码中加入 JavaScript 脚本,如下所示。目的是创建前台 asp 文件与后台 SQL Server 数据库的连接。

```
<%
Dim cnobj,strcon
set cnobj = Server.CreateObject("ADODB.Connection")
```

```
strcon = "Driver = {SQL Server};Server = localhost;DataBase = OnLinePaper;Uid = ;Pwd = "
cnobj.Open strcon
%>
```

3. 后台数据库插入、更新、显示、删除数据的代码(odbc_connection.asp)

数据库连接成功之后,在 odbc_connection.asp 文件中加入对后台数据库插入、更新、显示、删除数据的操作代码。这些操作都是数据库的基本操作,可参考 SQL 语句部分内容。

```
<! -- # Include File = "odbc_connection.asp" -->

<%
'检查对象是否存在函数
Function Check(SqlFind)
dim comm,rs
   Set comm = Server.CreateObject("ADODB.Command")
       comm.ActiveConnection = cnobj

       comm.CommandType = 1
       comm.CommandText = SqlFind
       set rs = comm.Execute
       if not rs.bof and not rs.eof  then
              cnobj.close
              set cnobj = Nothing
              Check = AdminName
       else
              Check = ""
       end if
End Function

'插入数据函数
Function Admin_Insert(InsertSql)
dim comm,rs

   Set comm = Server.CreateObject("ADODB.Command")
       comm.ActiveConnection = cnobj

       comm.CommandType = 1
       comm.CommandText = InsertSql
       comm.Execute
       Admin_Insert = "操作成功!"
End Function

'更新信息
Function Admin_Update(UpdateSql)
dim comm,rs

   Set comm = Server.CreateObject("ADODB.Command")
       comm.ActiveConnection = cnobj
```

```
        comm.CommandType = 1
        comm.CommandText = UpdateSql
        comm.Execute
        Admin_Update = "操作成功!"
End Function

'显示数据

Sub AdminShow(SqlSelect)
    Set rs = Server.CreateObject("ADODB.Recordset")
    rs.open SqlSelect,cnobj,1
End Sub

'信息删除
Function Admin_Delete(DeleteSql)
dim comm,rs

   Set comm = Server.CreateObject("ADODB.Command")
        comm.ActiveConnection = cnobj

        comm.CommandType = 1
        comm.CommandText = DeleteSql
        comm.Execute
        Admin_Delete = "删除成功!"
End Function
'
sub AdminShow1(SqlSelect)
dim comm
   Set comm = Server.CreateObject("ADODB.Command")
        comm.ActiveConnection = cnobj

        comm.CommandType = 1
        comm.CommandText = SqlSelect
        set rs_command = comm.Execute
end sub
%>
```

4. 教师出题代码(TeachCouChoose.asp)

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="936"%>
<!-- #Include File="AdminLoginSign.asp"-->
<!-- #Include File="../conn/AdminFunction.asp"-->
<HTML>
    <HEAD>
        <SCRIPT>
            function GetTeachName(id_class,id_course,type,value)
            {
            document.getElementById(id_class + id_course).innerHTML = "<SELECT name='" +
type + "'><%
            dim rs
```

```
                    call AdminShow("select * from TeachInfo")
                    do while not rs.eof
                      %><OPTION value = '<% = rs("TeachNum")%>'><% = rs("TeachName")%>
</OPTION><%
                         rs.movenext
                    loop
                     %></SELECT><INPUT type = 'hidden' name = 'teachClass' value = '" + id_class +
"'><INPUT  type = 'hidden' name = 'teachCourse' value = '" + id_course + "'><INPUT type =
'submit' value = '修改'>";

               }
          </SCRIPT>
          <META http-equiv = "Content-Type" content = "text/html; charset = gb2312" />
          <TITLE>授课操作</TITLE>
     </HEAD>
     <BODY style = "margin:0;text-align:center;">
      <%
      dim ClassNum,CouNum,TeachNum,InsertSign,rs_command,deleteclass,deletecourse,deleteSign,
teachClass,teachCourse,UpdateSign,UpdateTeachNum
            ClassNum = request("ClassNum")
            CouNum = request("CouNum")
            TeachNum = request("TeachNum")
            deleteclass = request("deleteclass")
            deletecourse = request("deletecourse")
            teachClass = request("teachClass")
            teachCourse = request("teachCourse")
            UpdateTeachNum = request("UpdateTeachNum")
            if teachClass <> "" and teachCourse <> "" and UpdateTeachNum <> "" then
                  UpdateSign = Admin_Update("update TeachCou set TeachNum = '" & UpdateTeachNum
& "'where ClassNum = '" & teachClass & "' and CouNum = '" & teachCourse & "'")
            end if
            '======================================================================
            if deletecourse <> "" and deleteclass <> "" then
                  deleteSign = Admin_Delete("delete from TeachCou where ClassNum = '" &
deleteclass & "' and CouNum = '" & deletecourse & "'")
            end if
            '======================================================================
            if ClassNum <> "" and CouNum <> "" and TeachNum <> "" then
                  call AdminShow1("select * from TeachCou where ClassNum = '" & ClassNum & "'and
CouNum = '" & CouNum & "'")
               if not rs_command.bof and not rs_command.eof then
                    InsertSign = "该班该专业已经选课"
               else
                      InsertSign = Admin_Insert("insert into TeachCou values('" & TeachNum & "','" &
CouNum & "','" & ClassNum & "')")
                    InsertSign = Admin_Insert("insert into PaperChoose(ClassNum,CouNum) values
('" & ClassNum & "','" & CouNum & "')")
               end if
            end if
            '=====================================================================
        %>
```

```
<TABLE width = "550" border = "0" cellspacing = "0" cellpadding = "0">
  <TR>
    <TD height = "60" align = "center" style = "font - size:18px;color:#990000;font - 
weight:800;">授课操作</TD>
  </TR>
  <TR>
    <TD height = "2" bgcolor = "#000000"></TD>
  </TR>
  <TR>
    <TD>
        <FORM name = "form1" method = "post" action = "">
        <TABLE width = "550" border = "0" cellspacing = "0" cellpadding = "0">
          <TR>
            <TD width = "170" height = "40">班级 </TD>
            <TD width = "140" height = "40">课程</TD>
            <TD width = "140" height = "40">教师</TD>
            <TD width = "100" height = "40">操作</TD>
          </TR>
          <%
          call AdminShow("select * from TeachCou")
          if request.QueryString("AbsoPage") <> "" then
              AbsoPage = cint(request.QueryString("AbsoPage"))
          else
              AbsoPage = 1
          end if
          if not rs.bof and not rs.eof then
              rs.pagesize = 13
              rs.absolutepage = AbsoPage
              dim i
              for i = 1 to rs.pagesize
                  if rs.eof then exit for
                  if i mod 2 = 0 then
                      response.write "<tr  bgcolor = '#f9ffe3'>"
                  else
                      response.Write "<tr bgcolor = '#fae8b7'>"
                  end if
          %>
      <TD height = 25 align = "left" valign = "middle"  style = "font - size;12px;">
          <%
              call AdminShow1("select * from ClassInfo where ClassNum = '" & rs("
ClassNum") & "'")
              if not rs_command.bof and not rs_command.eof then
                  response.write rs_command("ClassName")
              end if
          %>
      </TD>
      <TD align = "left" height = 25 style = "font - size;12px;">
          <%
              call AdminShow1("select * from CouInfo where CouNum = '" & rs("CouNum") & "'")
              if not rs_command.bof and not rs_command.eof then
                  response.write rs_command("CouName")
```

```
                end if
              %>
            </TD>
             < TD align = " left" height = 25 style = " font - size; 12px;" id = "<% = rs
("ClassNum") %><% = rs("CouNum") %>" >
                  < SPAN onClick = " GetTeachName ( '<% = rs ( " ClassNum") % >', '<% = rs
("CouNum") %>', 'UpdateTeachNum', '<% = rs("TeachNum") %>')">
                <%
                    call AdminShow1("select * from TeachInfo where TeachNum = '" & rs
("TeachNum") & "'")
                    if not rs_command.bof and not rs_command.eof then
                       response.write rs_command("TeachName")
                    end if
                 %>
              </SPAN>
            </TD>
            < TD >
                  < A HREF = " TeachCouChoose. asp? deleteclass = <% = rs ( " ClassNum") % >
&deletecourse = <% = rs("CouNum") %>" target = "_self">删除</A >
            </TD>
              <%
                    rs.movenext
                next
              %>
              < TR >
                < TD colspan = "4" align = "center">
                <%
              Response. Write "< A HREF = 'TeachCouChoose. asp?AbsoPage = 1' target = '_self'
style = 'font - size:13px;text - decoration:none;color:#660000;'>【首页】</A>   "   '
                if AbsoPage > 1 then
                         Response. Write "< A HREF = 'TeachCouChoose. asp? AbsoPage = " &
AbsoPage - 1 & "' target = '_self'  style = 'font - size:13px; text - decoration:none; color: #
660000; '>【上一页】</A>   "
                else
                    Response. Write "< SPAN style = 'font - size:13px; text - decoration:
none;color: # 660000; '>【上一页】</SPAN>   "
                end If
                if AbsoPage < rs. PageCount and AbsoPage <> rs. PageCount then
                         Response. Write "< A HREF = 'TeachCouChoose. asp? AbsoPage = " &
AbsoPage  + 1 & "' target = '_self'  style = 'font - size:13px; text - decoration:none; color: #
660000; '>【下一页】</A>   "
                else
                    Response. Write "< SPAN style = 'font - size:13px; text - decoration:
none;color: # 660000; '>【下一页】</SPAN>   "
                end If
                  Response. Write "< A HREF = 'TeachCouChoose. asp? AbsoPage = " & rs.
PageCount & "' target = '_self'  style = 'font - size: 13px; text - decoration: none; color: #
660000; '>【末页】</A>"
                   response. write "     < SPAN style = 'font - size: 13px; text -
decoration:none;color: # 660000;'>" & AbsoPage  & " / " & rs. pagecount & "</SPAN>"
            else
```

```
              response.write "还没有授课信息生成"
          end If
        %></TD>
              </TR>
            </TABLE>
          </FORM>
        </TD>
      </TR>
      <TR>
        <TD><FORM action="" method="get">
            <TABLE width="550" border="0" cellpadding="0" cellspacing="0" bgcolor="
#CCFFFF">
            <TR>
              <TD height="2" colspan="2" bgcolor="#990000"></TD>
            </TR>
            <TR>
              <TD width="150" height="30" align="center">班级：</TD>
              <TD width="400" align="center">
                <SELECT name="ClassNum" id="ClassNum" style="width:120px;">
                    <%
                  call AdminShow("select * from ClassInfo")
                  do while not rs.eof
                   %>
                      <OPTION value='<% =rs("ClassNum")%>'><% =rs("ClassName")%>
</OPTION>
                  <%
                      rs.movenext
                  loop
                   %>
                </SELECT>                    </TD>
            </TR>
            <TR>
              <TD height="30" align="center">课程：</TD>
              <TD height="30" align="center">
                <SELECT name="CouNum" id="CouNum" style="width:120px;">
                    <%
                  call AdminShow("select * from CouInfo")
                  do while not rs.eof
                   %>
                         <OPTION value='<% =rs("CouNum")%>' selected><% =rs("
CouName")%></OPTION>
                  <%
                      rs.movenext
                  loop
                   %>
                      </SELECT>                    </TD>
            </TR>
            <TR>
```

```
                        <TD height="30" align="center">教师:</TD>
                        <TD height="30" align="center">
                          <SELECT name="TeachNum" id="TeachNum" style="width:120px;">
                              <%
                            call AdminShow("select * from TeachInfo")
                            do while not rs.eof
                            %>
                                <OPTION value='<% =rs("TeachNum")%>'><% =rs("TeachName")%></
OPTION>
                            <%
                                rs.movenext
                            loop
                            %>
                          </SELECT>
                        </TD>
                      </TR>
                      <TR>
                        <TD height="30" colspan="2" align="center" bgcolor="#FFCCFF">
                          <INPUT type="submit" name="Submit" value="提交">

                                
                          <INPUT type="reset" name="Submit2" value="重置">          </TD>
                      </TR>
                    </TABLE>
                    </FORM>
                  </TD>
                </TR>
              </TABLE>

              <% =InsertSign%><% =deleteSign%><% =UpdateSign%>
              <%
              rs.Close
              Set rs = Nothing
              cnobj.close
              set cnobj = Nothing
               %>

          </BODY>
      </HTML>
```

10.4 本章小结

本章对 SQL Server 与 ASP 结合开发在线考试系统做了详细的介绍,从需求分析入手,进行系统的模块设计、数据库设计、程序设计,最后给出系统运行的源代码。

通过本章的学习,实现管理员、教师、学生 3 种不同权限的操作。管理员管理试卷信息和教师、学生信息。教师进行出题、阅卷、制定阅卷标准。学生在规定时间内进行考试,考试结束后自行查询考试成绩。通过本章学习,要求学生增强对数据库的设计分析的能力,熟练

利用 Dreamweaver 工具软件实现对在线考试系统的开发。

10.5 思考与练习

1. 在线考试系统数据库、表的创建方法有几种？具体操作步骤是什么？
2. 简述在线考试系统的功能。
3. 在线考试系统的阅卷代码是如何实现的？
4. 在线考试系统的试题添加、删除、修改代码是什么？
5. 完成在线考试系统代码测试并写出测试报告。

第11章 JSP案例——物资管理系统

随着计算机技术的高速发展,计算机在各个领域中的优势都不断体现出来,对于各个企业来说,之前所使用的以纸介质来记录物资的信息、交易记录、使用情况等物资管理方式存在着很多的弊端,当出现大量数据时容易出错,造成不必要的损失,所以已不能适应现代企业的发展。因此,实现物资管理的系统化、规范化、自动化已成为现代企业管理的必然趋势。

学习目标及要求

(1) 分析物资管理系统的需求。

(2) 能够进行物资管理系统的模块设计。

(3) 能够实现数据库设计。

(4) 利用工具软件,实现前台和后台数据库的连接。

(5) 实现每一个功能模块,并自行测试。

11.1 系统设计

本系统主要实现物资的出入库管理、信息的修改与统计,以及人员的注册、管理与交流。系统采用了B/S结构,Web浏览器是客户端最主要的应用软件,所以只要有网络,用户就可以随时更新消息,这样方便了操作人员,同时也能及时更新任何信息。本系统采用Sun公司的JSP语言开发,利用其“一次编写、到处运行”的特点,构造了一个可以在多个系统中运行的物资管理系统。

11.1.1 需求分析

物资管理对于一个企业来说是至关重要的。物资可以说是一个企业的命脉,所以物资进出的实效性就是关键所在。

物资管理系统的设计目标为:

(1) 为企业提供一个完善的出入库管理的在线平台,包括物资的查询、添加、修改。

(2) 员工和管理人员使用不同权限。

(3) 员工可以通过平台与管理人员进行交流。

11.1.2 开发运行环境

在现今软件开发模式中,主要有两种结构:C/S结构和B/S结构。C/S(Client/Server)

结构是大家熟知的客户机和服务器结构，采用开放模式，但只是系统开发一级的开放性，在特定的应用中无论是Client端还是Server端都还需要特定的软件支持。由于没能提供用户真正期望的开放环境，C/S结构需要针对不同的操作系统及不同版本的软件进行开发设计，加之产品的更新换代十分快，已经很难适应百台计算机以上局域网用户同时使用，而且代价高、效率低。B/S(Browser/Server)结构即浏览器和服务器结构，它是随着Internet技术的兴起，对C/S结构的一种变化或者改进的结构。在这种结构下，用户工作界面通过WWW浏览器来实现，极少部分事务逻辑在前端(Browser)实现，但是主要事务逻辑在服务器端(Server)实现，形成所谓三层结构。这样就大大简化了客户端计算机载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本(TCO)。而且B/S结构还具有维护和升级方式简单、成本低、选择更多等优点。所以本系统开发将选择B/S结构。

本程序的开发与运行环境如下。

- 操作系统：Windows XP Professional。
- 数据库管理系统：Microsoft Access 2003。
- 开发工具：Macromedia Dreamweaver。
- 服务运行：Apache Tomcat 6.0的服务器，JDK 1.6.0。

11.1.3 功能模块设计

物资管理系统是基于Web实现的，本系统主要围绕着物资的入库到出库的一系列操作进行，系统的功能模块也是基于此一一实现的。物资管理系统主要包括6个功能模块，如图11-1所示。

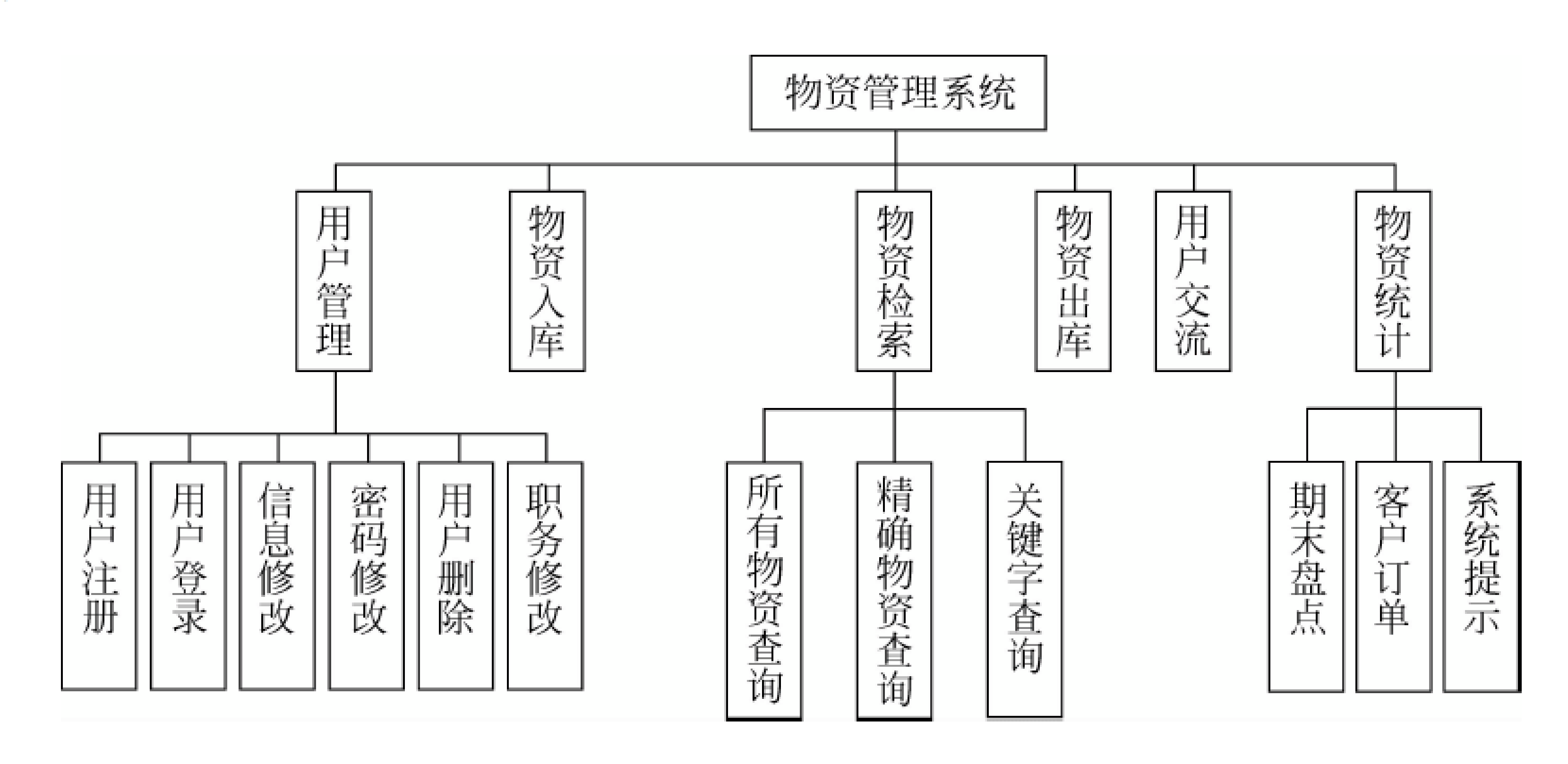


图11-1 系统模块图

1. 用户管理

用户管理主要包括用户注册、用户登录、信息修改、密码修改、用户删除、职务修改。

(1) 用户注册：用户注册时只能注册为普通人员，相应的权限也是最小的。

(2) 用户登录：用户通过输入正确的用户名、密码并且选择正确的职务，登录物资管理系统。

(3) 信息修改：信息修改可分为两部分，一个是用户对自己的信息的修改，另一个是管理员对他人信息的修改。

(4) 密码修改：管理员可以修改他人的密码，其他员工需要登录系统后通过此模块进

行修改。

(5) 用户删除：管理员可以删除长期未登录用户。

(6) 职务修改：管理员可以修改他人的职务，此模块只许管理员使用。

2. 物资入库

物资入库是管理员和入库人员共同完成的，管理员可以修改、删除物资信息，入库人员只能修改物资的数量。

3. 物资检索

物资检索可分为 3 种：一是所有物资查询；二是精确物资查询；三是关键字查询。

(1) 所有物资查询：当单击“搜索”按钮后，系统将会显示所有的物资。

(2) 精确物资查询：当知道物资的确切名称或者物资的确切价格时，可以通过这个模块快速找到所需的物资。

(3) 关键字查询：当知道物资名称的一部分时，可以通过这个模块进行查询。

4. 物资出库

物资出库为出库人员特有的模块，出库人员要填写订购数量和出库时间，出库时间不能少于系统时间，如果少于系统时间，那么系统将会报错。

5. 用户交流

用户交流模块是管理员和其他人员交流的平台，可以通过这个平台给管理员留言，其他人是看不见的。

6. 物资统计

物资统计模块包括期末盘点、客户订单、系统提示 3 个方面。

11.1.4 数据库设计

物资管理系统中设计了 6 个实体，分别是物资、管理员、入库人员、出库人员、普通员工和留言板，以及入库、出库、修改、查看等联系。在本系统中物资是核心，可以说这个系统都是围绕物资建立的。本系统 E-R 图如图 11-2 所示。

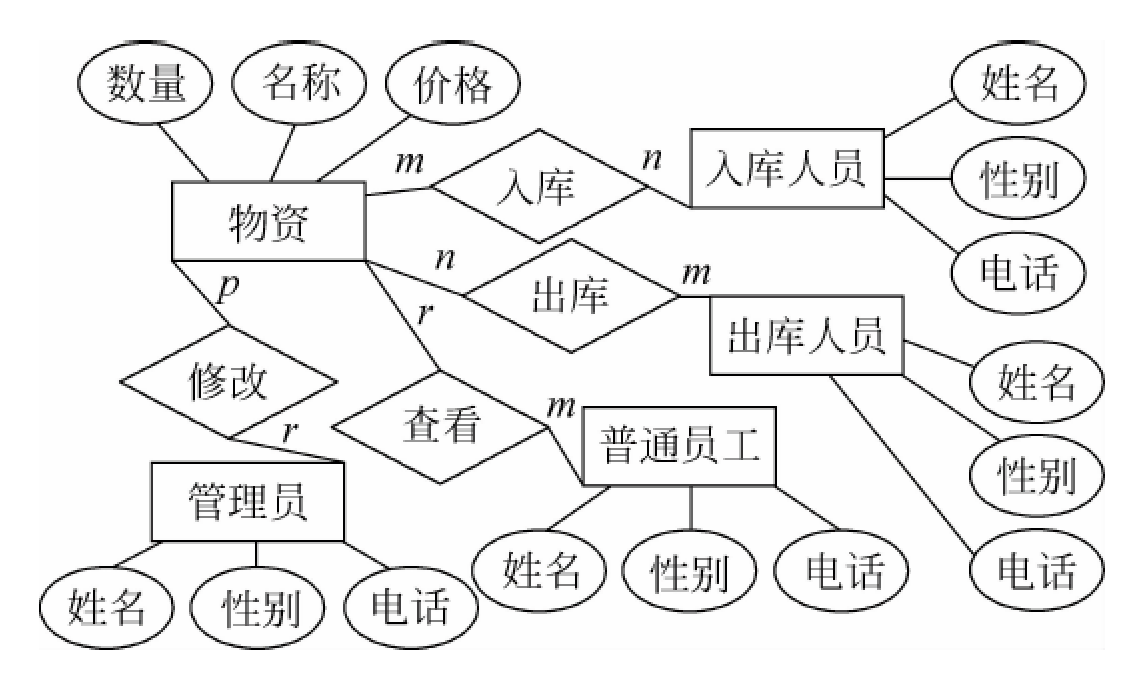


图 11-2 本系统整体 E-R 图

物资信息表如表 11-1 所示,记录了物资入库时所填写的信息,物资的名称和价格一般是不会改变的,入库数量与入库时间也是在新的同种物资入库时才改变的。

表 11-1　物资信息表

字　段　名	类　　型	说　　明
pid(主键)	数字	物资编号
pname	文本	物资名称
price	数字	物资价格
amount	数字	物资数量
date	日期/时间	入库时间

留言信息表如表 11-2 所示,不管是入库人员、出库人员还是普通人员都可以通过留言信息表与系统的管理员进行交流,也可以向管理员提出意见。当然除了管理员和留言者以外,其他人是无法查看留言信息的具体内容的。

表 11-2　留言信息表

字　段　名	类　　型	说　　明
id(主键)	文本	id 编号
name	文本	留言者姓名
theme	文本	留言主题
message	备注	留言内容

人员信息表如表 11-3 所示,主要记录在注册信息时所输入的个人信息。

表 11-3　人员信息表

字　段　名	类　　型	说　　明
id(主键)	文本	id 编号
name	文本	姓名
sex	文本	性别
email	文本	电子邮箱
tel	文本	电话号码
qq	文本	QQ 号码
passwd	文本	密码
birth	日期/时间	生日
zctime	文本	注册日期
zhiwu	数字	职务

出库物资信息表如表 11-4 所示,记录物资出库时需要的信息、出库时间和办理出库的人员等,通过对此表的查询就可以清楚地了解到物资出库所需要的所有信息。

表 11-4 出库物资信息表

字 段 名	类 型	说 明
number(主键)	文本	编号
username	文本	人员姓名
pid	数字	物资编号
pname	文本	物资名称
quantity	数字	出库数量
price_new	数字	新的物资价格
cost	数字	总价格
date	日期/时间	出库日期

11.2 系统实现

11.2.1 用户管理模块的实现

1. 用户登录

用户登录界面如图 11-3 所示。用户填写正确的用户名、密码,选择正确的职务,然后单击"提交"按钮,经系统判断正确后,页面将跳转到相对应的页面;单击"重置"按钮将清空所填的用户名和密码;单击"用户注册"将跳转到注册页面。如果填写内容不正确,系统将给予错误提示。

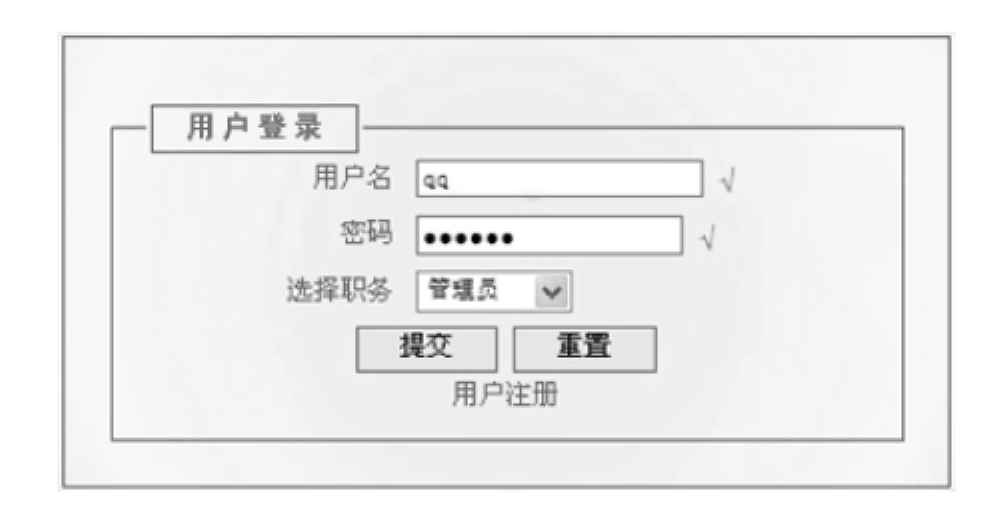


图 11-3 用户登录界面

本系统的密码限制为最少 6 位,加密方式采用 md5 加密模式,在数据库中我们所看到的密码那个字段也只是字符串,不是密码明文,这为系统的安全起到了保障。

在这个系统中,我们所输入的用户名可以是中文、英文,也可以是数字,但是它有个最少两个字符的限制。在输入英文和数字时一切显示正常,在输入中文时出现的是乱码,这需要转换。转换代码如下:

```
<%    public String getString(String s) {
        if (s == null) {
            s = "";
        }
        try {
            byte b[] = s.getBytes("ISO-8859-1");
```

```
            s = new String(b);
        } catch (Exception e) {
        }
        return s;
} %>
```

登录界面的主要任务是判断用户所输入的用户名、密码、职务是否正确，如果判断不出来或者明明是错的还能登录，那登录界面的意义就没有了，所以需要很严密的判断。判断代码如下：

```
if (!userpasswd.equals(info.getString("passwd"))) {
            out.println("<script language = \"javascript\">alert
(\"密码不正确...\");</script>");
//密码不正确，提示"密码不正确"信息
            out.println(" <script language = 'javascript'>");
            out.println("location = './login.jsp'; </script>");
                } else {
                if(!userzw.equals(info.getString("zhiwu"))) {
            out.println("<script language = \"javascript\">alert
   (\"用户选择职务不正确...\");</script>");
//职务不正确，提示"用户选择职务不正确"信息
        out.println(" <script language = 'javascript'>");
        out.println("location = './login.jsp'; </script>");}
            else if(Integer.parseInt(userzw) == 1)
   职务的判断
            { session.setAttribute("login", usernames);
        info.close();
        stmt.close();
        conn.close();}        //关闭所有对象
   catch (Exception e) {
   //用户名的判断
         out.println("<script language = \"javascript\">aler
t(\"用户名不正确...\");</script>");
//用户名不正确，提示"用户名不正确"信息
            out.println(" <script language = 'javascript'>");
            out.println("location = './login.jsp'; </script>");
        }
```

当用户所输入的用户名、密码、职务都正确时，将使用 session 对象将登录信息存入 Web 服务器的内存中。代码如下：

```
session.setAttribute("login", usernames);
session.setAttribute("id",id);
```

获取表单中的信息，用的是"request.getParameter("")"，在双引号中间的值为要获取表单的 name 值，当表单中所输入的内容为中文、密码时，都需要转换。转换代码如下：

```
String usernames = request.getParameter("username");          //获取表单内容
usernames = getString(usernames);                             //为中文的转换
String userpasswd = request.getParameter("userpassword");
String userzw = request.getParameter("userzhiwu");            //为密码的转换
```

连接数据库是系统中最为重要的一部分，因为差不多在每个页面都需要连接数据库。连接数据库的代码如下：

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");                    //连接数据库的驱动
Connection conn = DriverManager.getConnection("jdbc:odbc:wuzi");  //数据源的使用
Statement stmt = conn.createStatement();                          //接收变量
info.close();        //关闭对象
stmt.close();        //关闭对象
conn.close();        //关闭对象
```

2. 用户注册

用户注册界面如图 11-4 所示。用户注册界面可以判断用户名、邮箱是否被注册，内容是否为空，格式是否正确，如图 11-5 所示。

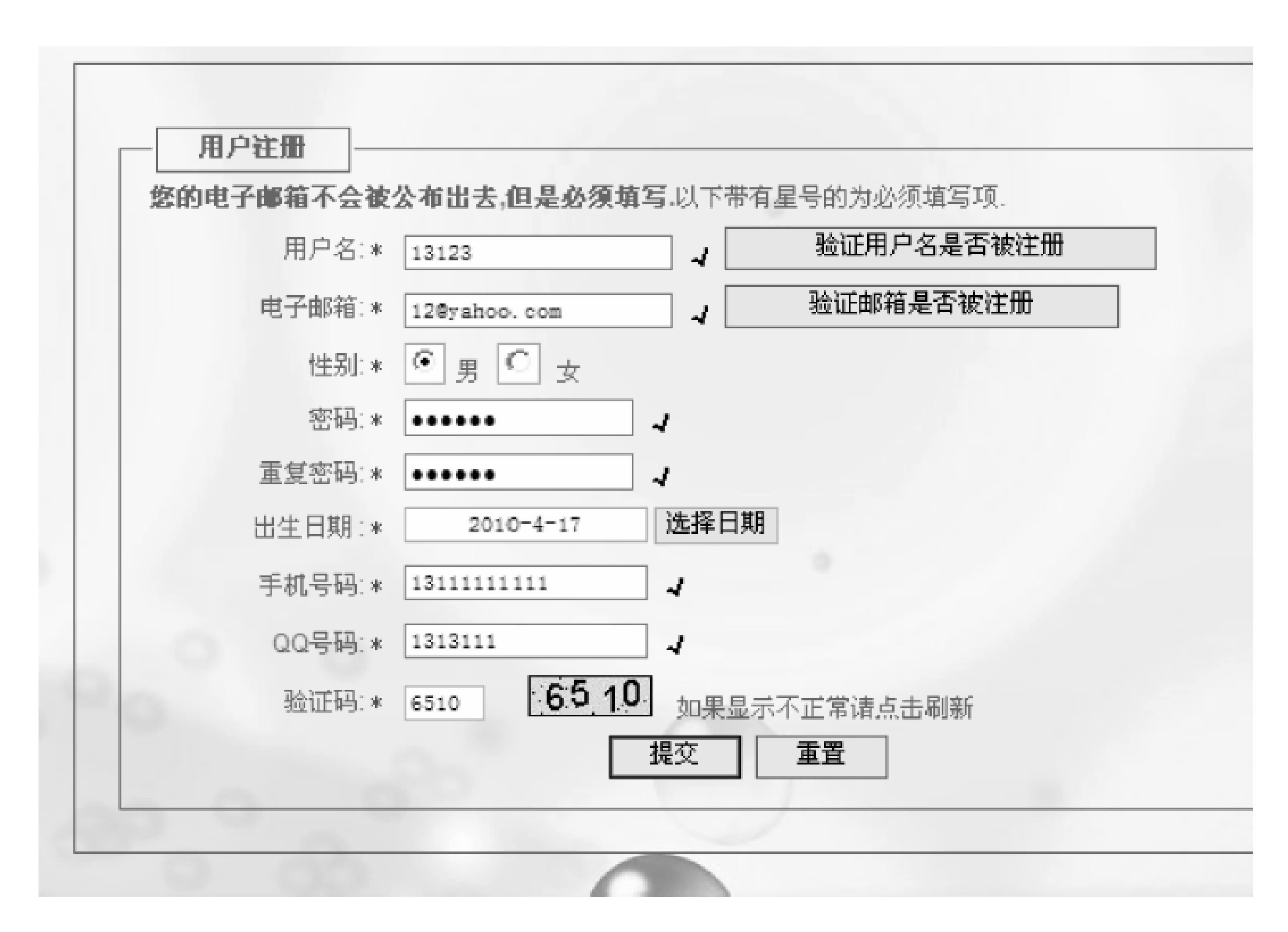


图 11-4 用户注册界面



图 11-5 判断界面

用户注册界面分为两个页面：一个是 regedit.jsp，主要面向用户；一个是 useradd.jsp，主要是后台操作、连接数据库。用户注册时在 regedit.jsp 的表单中所填入的信息将被传送到 useradd.jsp 页面，然后存入数据库。

出生日期可以直接双击表单填写，也可以单击右侧的“选择日期”按钮，在弹出的页面中进行选择，如图 11-6 所示。用户注册页面没有选择职务这一项，因为在这里注册的都是权限最低的普通人员，只具有查询物资和给管理员留信息两个功能，如果想要改变自己的权限只能给管理员留言，通过管理员来修改。这个注册页面还为防止某些人恶意注册特意添加了验证码这一项，验证码是通过 chknumber.jsp 页面来实现的。

图 11-6 选择日期

useradd.jsp 页面用来连接注册页面和数据库，在实际操作中这个页面是不会显示的，当我们所填的内容准确无误时，它将会把我们所填的内容存储到数据库的 user 表中，并跳转到登录页面；如果我们所填的内容出错时，它将会跳转回注册页面，使用户重新注册。

3. 管理员主界面

管理员主界面有两个功能：一是管理其他用户；二是管理物资（管理物资将在物资管理模块中讲解）。管理其他用户主要是修改用户的信息、权限和删除用户等。修改用户的信息时，管理员可以修改用户的性别、密码、出生日期、手机号、QQ 号和用户的职务等。在进入修改信息页面时，系统会连接数据库读取被修改人的信息（除密码外），如果密码不填写，密码将不予改动。修改信息界面如图 11-7 所示。



图 11-7 修改信息

修改信息应该使用 update。主要代码如下：

```
String sql = "update user set sex = '" + usersex + "',birth = '" + userbirthtime + "',qq = '" +
userqq + "',zhiwu = '" + userzhiwu + "', passwd = '" + userpassword + "',tel = '" + userphone +
```

```
"'where id=" + userid;
//更新数据
  int req = stmt.executeUpdate(sql);
  if (req > 0)                                         //查看数据是否更新成功
```

删除用户信息通过 usersdel.jsp 页面实现，这个页面也不在实际操作中显示，当管理员单击“删除”按钮后，此用户将被直接删除。主要代码如下：

```
try {
        String[] users = request.getParameterValues("users");
        for (int i = 0; i < users.length; i++) {            //循环删除多行记录
    stmt.executeUpdate("delete from user where id = "
            + users[i]);
        }
    } catch (Exception e) {
    }
} else if (action.equals("onlydel")) {                      //删除单条记录
    String userid = request.getParameter("id");
    stmt.executeUpdate("delete from user where id = " + userid);
}
```

4. 个人修改信息界面

个人修改信息同管理员修改并无太大区别，只是没有修改职务一项。

11.2.2 物资检索模块的实现

物资检索应用于此系统的入库界面、出库界面、管理员界面和普通人员界面，这个系统中的所有人都离不开这个物资检索模块。

1. 所有物资的查询

所有物资的查询顾名思义就是可以查询所有物资，如图 11-8 所示。可以查询到物资的名称、单个价格和数量。在查询结果页面一页只显示 5 种物资，当然也可以通过修改代码增加显示的数量，并且可以通过单击“第一页”、“上一页”、“下一页”、“最后一页”来翻页，在页面右侧显示了所在页数、总页数、记录数，还可以通过在表单中输入具体数字来跳转到指定页数。

第一页 上一页 下一页 最后一页 转到第: 页 GO 第1页 共

编号:1	品名:bike	价格:20
编号:2	品名:car	价格:1000
编号:3	品名:cup	价格:12
编号:5	品名:goodcar	价格:2000
编号:6	品名:T-shit	价格:21

图 11-8 所有物资的查询

主要代码如下：

```
int intPageSize;                //一页显示的记录数
int intRowCount;                //记录总数
int intPageCount;               //总页数
int intPage;                    //待显示页码
String strPage;
int i,j,k;
<% i=0;
  while(rs.next()){i++;}
  intRowCount = i;
//按行数计算总页数
  intPageCount = (intRowCount+intPageSize-1) / intPageSize;
  //调整待显示的页码
if(intPage>intPageCount){intPage = intPageCount;} %>
<TABLE align="center" width="100%">
<TR><TD align="center">
<% if(intPage>1){ %>
<A HREF="userorder.jsp?page=1">第一页</A><% }else
out.println("第一页");
//如果去掉这句,当没有“上一页”时就不显示出来了 %>
<TD align="center">
转到第:<INPUT type="text" name="page" size="8"> 页
<SPAN><INPUT  tySpe="submit" value="GO" name="cndok"></SPAN></TD>
<TD align="center">
第<FONT color="#993399"><% = intPage %></FONT>页</TD>
<TD align="center">
共<FONT color="#9900CC"><% = intPageCount %></FONT>页</TD>
<TD align="center">
共<FONT color="#990066"><% = intRowCount %></FONT>条记录</TD>
```

2. 精确物资查询

精确物资查询时，必须知道物资的具体名称或者物资的具体价格，只要知道其中一样就可以查询了。

主要代码如下：

```
if(pnamepname!=null&&(!pnamepname.equals(""))
&&priceprice!=0){
      sql="select * from product where pname=
'"+pnamepname+"' and price="+priceprice;}
  else if((pnamepname==null||pnamepname.equals(""))
&&(priceprice!=0)){
      sql="select *  from product where price="+priceprice; }
  else{
      sql="select *  from product where pname='"+pnamepname+"'";
}
```

3. 关键字查找

关键字查找时，不必知道物资全名，可进行模糊查找，主要代码如下：

```
  rs = sm.executeQuery(sql);
sql = "select *  from product where pname like '%" + pna + "%'";
```

11.2.3 物资入库模块的实现

物资入库主要是管理员和入库人员共同完成的，不同之处在于管理员可以随时修改物资的信息，但是入库人员只能在新物资入库的时候填写而且不能修改。

1. 新物资入库

新物资入库由入库人员填写，界面如图 11-9 所示。

入库单

产品名称: 44111
产品价格: 1312 [单位:元]
入库数量: 313
入库日期: 2010-04-17
确定 取消

图 11-9 入库单界面

2. 已有物资查询

管理员和入库人员对已有物资都有操作的权利，管理员进行修改，入库人员填写物资增加的数量。

修改信息：

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver").newInstance();
String url = "jdbc:odbc:wuzi";
String user = "";
String pas = "";
con = DriverManager.getConnection(url, user, pas);
sm = con.createStatement();
rs = sm.executeQuery("select * from product where pname = '" + pname + "'");
while(rs.next()){
}String condition1 = "update product set amount = '" + amount + "',
  price = '" + priceq + "'  where  pname = '" + pname + "'";//数量更新
  sm.executeUpdate(condition1);
```

增加物资：

```
amount = amoun + amount;
      String condition1 = "update product set
amount = '" + amount + "', price = '" + priceq + "'  where  pname = '" + pname + "'";
```

11.2.4 物资出库模块的实现

物资出库为出库人员特有的模块，物资出库单中只须填写订购的数量和交货的日期即

可。需要注意的是：交货日期一定要比当天时间晚，如果小于当天日期系统将会出现“预交货期小于当前日期，请重新下订单”的提示；此外，订购数量尽量不要大于存货数量，以免物资来不及补充；当出库的总数达到一定金额时将可以享受打折的优惠。下订单页面如图 11-10 所示。

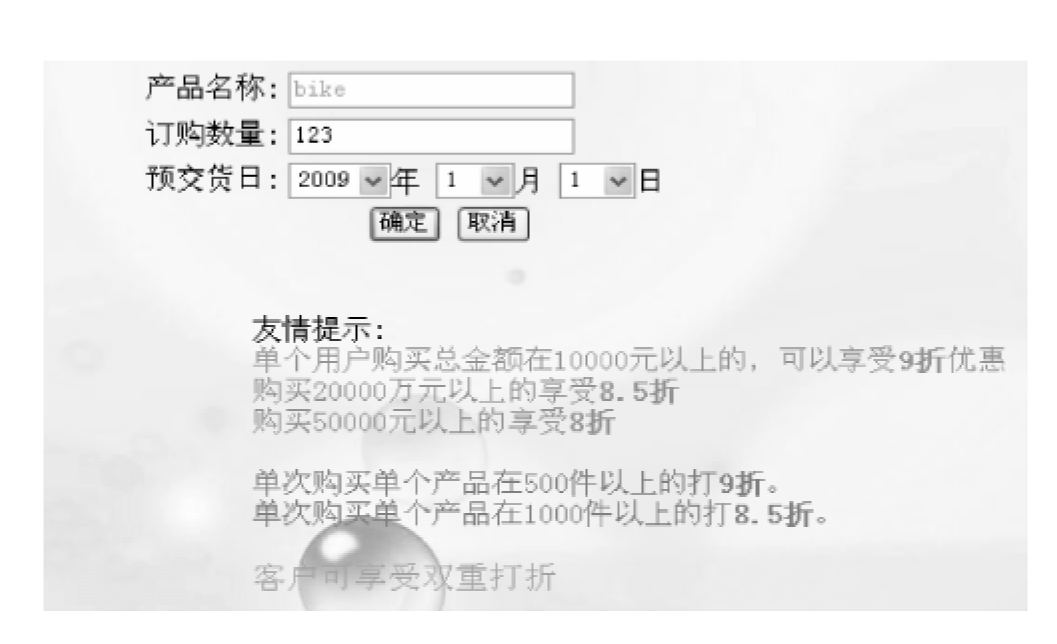


图 11-10 下订单页面

当仓库中物资出库时，出库人员只须查找出库单，并删除此次的出库记录，使库存数量能够得到及时更新。

11.2.5 物资统计模块的实现

物资统计模块主要包括期末盘点、客户订单和系统提示。

1. 期末盘点

期末盘点可以查询从一段时间到另一段时间的各个物资的库存量，可以通过期末盘点了解在一段时间内的某些物资的入库出库情况，便于提前考虑增加或者减少物资的入库情况。期末盘点界面如图 11-11 所示。

期末盘存，从 2009 年 1 月 1 日 到 2009 年 1 月 1 日 提交
您正在查询的库存量是从2009-01-01到2011-01-01的量
编号：1 产品名称：bike 产品价格:20 产品数量:1010 入库时间:2009-05-09 00:00:00
编号：2 产品名称：car 产品价格:1000 产品数量:12 入库时间:2009-05-25 00:00:00
编号：3 产品名称：cup 产品价格:12 产品数量:1000 入库时间:2009-05-25 00:00:00
编号：5 产品名称：goodcar 产品价格:2000 产品数量:12 入库时间:2009-05-28 00:00:00
编号：6 产品名称：T-shit 产品价格:21 产品数量:4000 入库时间:2009-05-28 00:00:00
编号：7 产品名称：english-book 产品价格:12 产品数量:200 入库时间:2009-05-28 00:00:00
编号：8 产品名称：chinese-book 产品价格:56 产品数量:699 入库时间:2009-05-28 00:00:00
编号：9 产品名称：computerbook 产品价格:89 产品数量:777 入库时间:2009-05-28 00:00:00

图 11-11 期末盘点

查询代码如下：

```
        rs = sm.executeQuery("select count( * ) from product");
        while(rs.next()){count = rs.getInt(1); }
rs = sm.executeQuery("select * from product");
        //仓库查询时,可以使用 select * from product where date > 2009 - 5 - 7
        while(rs.next()){pidd = rs.getInt(1); pname = rs.getString(2); price = rs.getInt(3);
```

```
                amount = rs.getInt(4); date = rs.getString(5);
if(date.compareTo(C)<= 0&&date.compareTo(B)>= 0){
//如果入库时间大于盘点查询输入的时间,则输出编号、名称、产品价格、数量、入库时间
                out.println("<tr><td>编号: " + pidd + "</td>");
                out.println("<td>产品名称: " + pname + "</td>");
out.println("<td>产品价格:<span style = 'color:#CC0033'>" + price + "</span></td>");
out.println("<td>产品数量:<span style = 'color:#CC0066'>" + amount + "</span></td>");
out.println("<td>入库时间:<span style = 'color:#CC00FF'>" + date + "</span></td></tr>");
                     out.println("<tr><td colspan = '5'>
```

2. 客户订单

客户订单主要表现在对出库人员的出库物资的统计,并显示出库物资的总金额。客户订单界面如图 11-12 所示。



图 11-12 客户订单

11.2.6 其他模块的实现

1. 欢迎用户

欢迎用户是通过在登录信息时保存在 session 对象中的用户名实现的,代码如下:

```
String name = (String) session.getAttribute("login");
```

2. 最新物资提示

最新物资提示就是查询在 product 中前 5 位的物资名称,界面如图 11-13 所示。

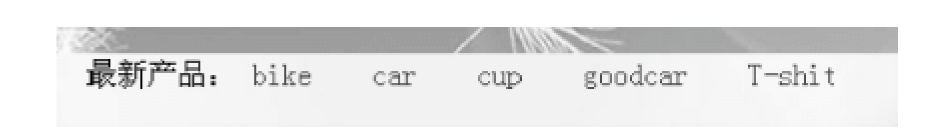


图 11-13 最新物资提示

3. 用户交流模块

用户交流就是一个留言板,对于入库人员、出库人员、普通员工,此留言板都是可以留言的,界面如图 11-14 所示。



图 11-14 留言板

主要代码如下：

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection conn = DriverManager.getConnection("jdbc:odbc:wuzi"); //数据源的使用
Statement stmt = conn.createStatement();
String sql = "insert into lyb(name,theme,message) values('"
             + username + "','" + theme + "','" + message + "')";
int req = stmt.executeUpdate(sql);
if (req == 1) {out.println("更新成功");
    out.println("<script language='javascript'>");
    out.println("location='./uusearch.jsp';</script>");
} else {out.println("更新失败");
    out.println("<script language='javascript'>");
    out.println("location='./lyb.jsp';</script>");
}stmt.close()conn.close();
```

只有管理员可以查看留言板，留言板列表中有留言者的姓名、留言主题和删除选项，当单击留言板上的姓名后可以查看留言者、留言主题、留言内容。留言详细内容如图 11-15 所示。

编号：2
用户名称:11
留言主题:
留言内容:
返回上一页

图 11-15 留言详细内容

删除留言的主要代码如下：

```
psm = con.prepareStatement("delete from lyb where id = ?");
psm.setInt(1,ppid);
  psm.executeUpdate();
response.sendRedirect("showlyb.jsp");
```

11.3 系统运行

1. 登录测试

为了确保物资管理系统的安全性，用户必须登录才可以使用本系统。当用户输入错误的用户名、密码或者职务时，将不能登录系统，并且有错误提示，如图 11-16 所示。

当用户输入正确的用户名、密码和职务时，系统将会根据用户所选择的职务跳转到不同

的页面中去。

2. 出库测试

1）增加订单测试

增加订单时要求产品名称是用户之前选择的，不能更改，用户只须输入正确的订购数量，选择正确的交货日期即可，如图 11-17 所示。



图 11-16 出错提示

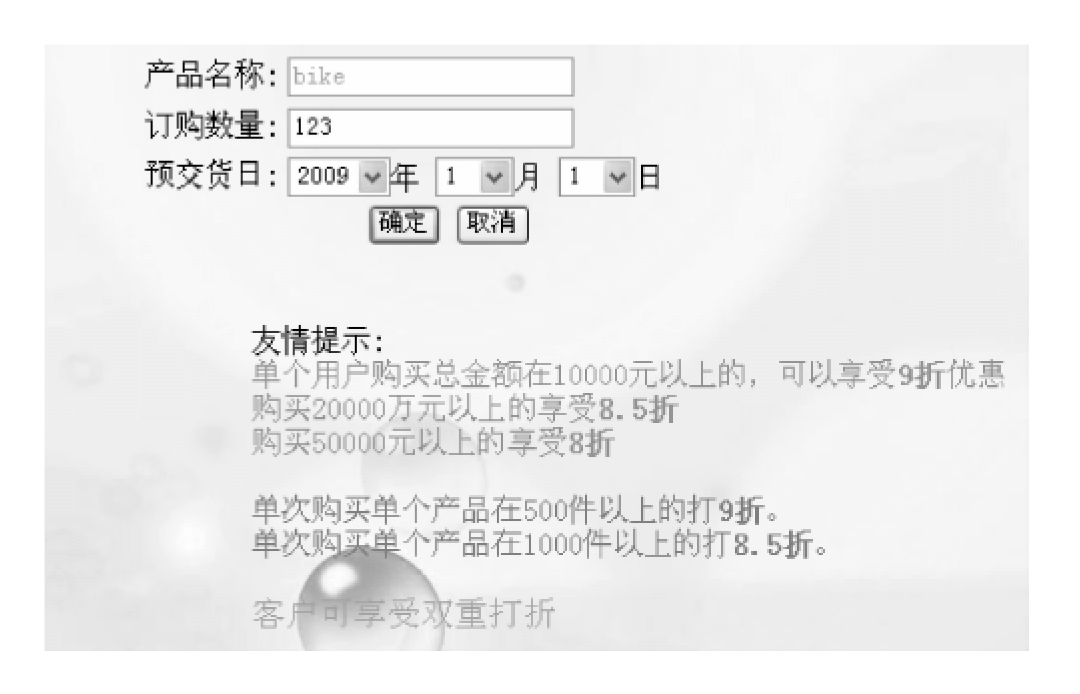


图 11-17 增加订单

增加订单时如果交货日期小于当天日期，将会出现错误提示，如图 11-18 所示。



图 11-18 日期出错

2）删除订单测试

删除订单时，只须找到想要删除的订单，单击“删除”按钮即可，当单击“删除”按钮后，系统会将删除订单中的物资数量加回到库存数量中。

11.4 本章小结

本系统是一个具有一定实用性的数据库信息管理系统。它主要完成对物资仓库的一系列管理，包括入库、出库以及员工和留言板信息的管理。本系统采用目前 Web 开发技术中应用最广泛的 JSP 来完成整个系统的设计，在数据库的设计上采用了简单、便利和实用的关系数据库 Access。

在系统的功能实现方面具有操作简单、方便实用的特点。在用户管理方面，实现了用户注册、密码修改和个人信息修改等各项功能。在物资入库、出库时对物资信息的修改等方面通过设计、使用也很好地展现出来，并且通过系统提示和期末盘点等功能使操作人员更好地管理物资。在查询方面，系统具有3种查询方式，使人员在查询时更加方便。

11.5 思考与练习

1. 简述JSP对数据库的操作步骤。
2. 简述物资信息表、留言信息表、人员信息表的创建代码。
3. 简述物资管理系统的功能。
4. 完成物资系统的入库功能。
5. 实现物资系统的出库功能。
6. 制作用户留言板的界面和实现留言的功能。

第四篇

新技术与网站管理

第12章 XML及其应用

标记行为由来已久，比如我们所熟知的对财务报表的注释、差旅费用的名细等，都可以理解为标记。标记的意思就是为某物加上记号(marking it up)，现在有很多的文本处理系统也正是基于这个原理而产生的。其中由标记引发而来的标记语言的杰出代表之一就是XML。

学习目标及要求

(1) 了解XML的发展。

(2) 理解XML的术语。

(3) 了解XML的语法。

(4) 理解并讨论XML和HTML有何不同。

(5) 能够使用层叠样式表相关联的XML文件。

12.1 XML概述

XML是extensible markup language的缩写，是由万维网联盟(World Wide Web Consortium，W3C)定义的一种语言，称为可扩展标记语言。所谓可扩展性是指XML允许用户按照XML规则自定义标记。XML文件是由标记及其所标记的内容构成的文本文件，与HTML文件不同的是，这些标记可自由定义，其目的是使得XML文件能够很好地体现数据的结构和含义。W3C推出XML的主要目的是使得Internet上的数据相互交流更方便，让文件的内容更加通俗易懂。

一个简单的XML文件：

```
<?xml  version = "1.0"  ?>
<职员>
    <姓名>
          张小三
          <性别> 男  </性别>
          <出生日期>  1980 年 8 月 15 日 </出生日期>
    </姓名>
</职员>
```

上面的XML文件包含一个XML声明<? xml version＝"1.0" ? >和4个标记。每个标记都必须包括开始标记和结束标记，开始标记和结束标记之间的内容称为该标记所标

记的内容，简称“标记的内容”。一个标记的内容中可以包含文本或其他的标记，其中包含的标记称为该标记的子标记。XML 文件有且只有一个根标记，其他标记都必须封装在根标记中，文件的标记必须形成树状结构。上面的 XML 文件的根标记的开始标记是“<职员>”，结束标记是“</职员>”，该根标记有一个子标记“<姓名>…</姓名>”，而标记“<姓名>…</姓名>”的子标记是“<性别>…</性别>”和“<出生日期>…</出生日期>”。XML 文件必须符合一定的语法规则，只有符合这些规则，XML 文件才可以被 XML 解析器解析。后续的章节会详细地讲解 XML 的语法规则。

下面的 XML 文件都是错误的，其中的“A. xml”没有根标记，“B. xml”标记没有形成树状结构(标记有交叉)。

(1) A. xml：

```
<?xml  version = "1.0"  ?>
<名称> 电视机  </名称>
<价钱> 3078 元 </价钱>
```

(2) B. xml：

```
<?xml  version = "1.0"  ?>
<张小林>
        <性别>  男  <出生日期>
        </性别> 1980 年 8 月 6 日 </出生日期>
</张小林>
```

12.2 XML 基础

12.2.1 XML 术语

在学习 XML 之前，先给大家介绍一些 XML 使用的主要术语。

元素(elements)：代表 XML 文档中的每一个组件，包含字符、元素或者两者皆有，允许有空元素(empty element)。

属性(attributes)：放在元素后面的数据，代表这个元素的另一种特性和特征。

统一编码(unicode)：一种字符集，包括全世界各种语言中有用的字符。

实体：“文本片”，实体也像一个缩写，一个实体可以是一个字符也可以是一个文档的数据，是 XML 文本的基本单位。

标记(markup)：像标记(TAG)、实体引用(entity reference)和声明这样的构件。

样式表(XSL)：描述 XML 的元数据文件格式的语言(因为 CSS 对于 XML 来讲不够用)。

序言(prolog)：关于文档实例表现形式的信息。

文档实例(document instance)：以元素的层次结构组织的实际文档数据。

12.2.2 XML 语法

1. 一个简单的例子

下面是一个关于库存物品清单的简单例子，大家可以从中发现 XML 的许多特性，并且

可以体会到XML的元数据文件是非常清晰易读的。

```
<! - XML 文件示例 -- >
<?xml version = "1.0" encoding = "GB2312" ?>
<库存清单>
 <! -- 这里就是注释 -- >
 <库存物品>
 <名称>毛巾</名称>
 <规格 单位 = "cm">25 * 50 </规格>
 <生产厂家>中国棉纺厂</生产厂家>
 <生产日期>2010 - 09 - 01 </生产日期>
 <数量 单位 = "条">400 </数量>
 <进货价 单位 = "元">2.5 </进货价>
 <零售价 单位 = "元">6.5 </零售价>
 </库存物品>
 <库存物品>
 <名称>碗</名称>
 <规格 单位 = "cm">5 * 15 </规格>
 <生产厂家>中国搪瓷厂</生产厂家>
 <生产日期>2010 - 09 - 11 </生产日期>
 <数量 单位 = "个">200 </数量>
 <进货价 单位 = "元">1.5 </进货价>
 <零售价 单位 = "元">4.5 </零售价>
 </库存物品>
 <库存物品>
 <名称>筷子</名称>
 <规格 单位 = "个">1 * 10 </规格>
 <生产厂家>中国搪瓷厂</生产厂家>
 <生产日期>2010 - 09 - 11 </生产日期>
 <数量 单位 = "把">200 </数量>
 <进货价 单位 = "元">1.5 </进货价>
 <零售价 单位 = "元">3.5 </零售价>
 </库存物品>
</库存清单>
```

看过了XML语言的基本格式，下面将详细讲解XML元数据文件的各个组成部分。实际上，就上面这些而言是非常简单的，当然，要想更清楚地了解和学习XML，就必须清楚地知道这些定义的意义。

2．声明(declaration)

XML数据文件的定义不像HTML那样自由，它是非常严格的，缺少一个就会出错。因为每个XML的解析器第一步要做的事情就是检查XML文件是否合格，就是“well-formed”，所以大家看到第一句是这样的：

```
<?xml version = "1.0" encoding = "GB2312" ?>
```

注意：不能漏掉此句，否则解析器将不予解析。当然里面的encoding可以不用写，默认

是 Unicode。Unicode 编码结合 ISO 的字库标准，可以满足非英语语系国家的需要。

3. 元素(elements)和属性(attributes)

```
<元素名 属性名 = "属性值">
    <子元素>元素的数据</子元素>
      </元素名>
```

看上面的这个例子，就能很形象地理解什么是元素、什么是属性，但是需要注意的是元素是区分大小写的，也就是说，<t></t>和<T></T>不是同一个元素。还有就是元素里的数据只要是合法的，就会完全忠实地成为数据，例如：

```
<元素名>这是一个</元素名>
<! - 这是另一个 -->
<元素名>这是一个 </元素名>
<! - 这又是另一个 -->
<元素名>这是一个
</元素名>
```

第一个数据、第二个数据只差一个空格，第三个数据则多了一个回车符，这 3 个数据是不同的数据，因此在写这些数据时一定要小心。

XML 中的属性值必须使用单引号或双引号括起来。

4. 注释(comments)

注释是在文档中嵌入关于文档和标记的有关信息，可以帮助别人和自己清楚代码的内容。

在 XML 中的注释是这样的：

```
<! - 啊，这就是帮助 —>
```

在这个注释中间，可以放入除了“--”符号以外的所有字符。

12.3 XML 和 HTML 有何不同

HTML 是用来编写 Web 页的语言，HTML 把数据和数据的显示外观捆绑在一起，如果只想使用数据而无须显示外观，可以想象，将数据和外观分离是多么的困难。HTML 不允许用户自定义标记，目前的 HTML 大约有 100 多个标记。HTML 不能体现数据的结构，只能描述数据的显示格式。下列 HTML 将数据分别用黑体 1(H1)、黑体 2(H2)和黑体 3(H3)来显示：

```
<HTML>
    <H1>
          张小三
          <H2>  男  </H2>
          <H3>  1980 年 8 月 15 日  </H3>
     </H1>
```

```
    <H1>
          王小妹
         <H2>女   </H2>
       <H3>  1985年12月7日   </H3>
    </H1>
</HTML>
```

上述HTML从文档结构上无法体现数据“张小三”和“1980年8月15日”之间的逻辑关系，只能将HTML中标记“<H1>…</H1>”所标记的文本内容用黑体显示在浏览器中。

与HTML不同的是，XML可以很好地描述数据的结构，有效地分离数据的结构和表示，可以作为数据交换的标准格式。XML可以自定义标记，标记名称是对所标记的数据内容含义的抽象，而不是数据的显示格式，XML只关心数据的组织结构，以便XML解析器按照其结构分解出数据，XML本身不提供数据的显示格式。浏览器不能直接显示XML文件的标记内容，如果想要浏览器显示XML文件中标记的内容，就必须以某种方式告诉浏览器如何显示，一种方式是使用层叠样式表(CCS)，另一种方式是使用XSL变换。

以下的XML文件将自己关联到一个层叠样式表，以便浏览器能显示XML文件中标记所标记的内容。层叠样式表中最重要的组成部分就是样式表，其作用是说明XML文件中的标记内容用何种方式来显示。比如，如果想用黑体来显示XML文件中标记“<name>…</name>”所标记的文本内容，就可以在层叠样式表文件中包含如下的样式表：

```
name
{
display:block;font-size:36pt;font-weight:bold
}
```

该样式表的作用是告知浏览器将标记“<name>…</name>所标记的内容显示在一个“块区域”中。

```
birthday
{
display:line;font-size:12pt;font-weight:bold
}
```

该样式表的作用是告知浏览器将标记“<birthday>…</ birthday>”所标记的内容显示在一行中。

当使用层叠样式表和XML关联时，XML文档中的标记的名字不要含有非ANSI字符。

以下的show.css是一个简单的层叠样式表，该文件可以保存编码为ANSI的文件，并和其关联的XML文件存放在同一目录中。

```
show.css
  name
  {
  display:block;font-size:18pt;font-weight:bold
```

```
}
sex
{
display:line;font - size:16pt;font - weight:bold
}
birthday
{
display:line;font - size:9pt;font - weight:bold
}
```

XML 文件使用操作指令

```
<?xml - stylesheet  ?>
```

关联到某个层叠样式表，例如：

```
<?xml - stylesheet  href = "show.css"  type = "text/css"  ?>
```

以下是一个与 show. css 层叠样式表相关联的 XML 文件，用浏览器打开 Cha1_1. xml 文件，浏览器就会按照 show. css 中的样式表来显示 XML 文件中标记所标记的文本内容，效果如图 12-1 所示。

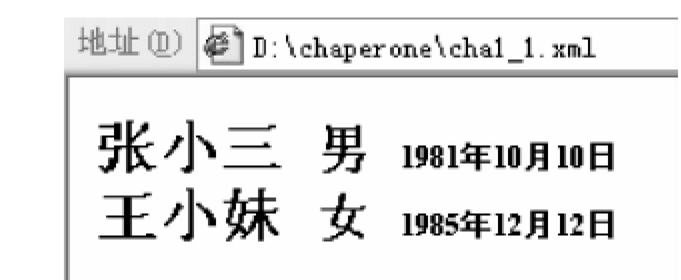


图 12-1 使用 CSS 显示 XML 中的数据

```
Cha1_1.xml
<?xml  version = "1.0"  ?>
<?xml - stylesheet href = "show.css" type = "text/css" ?>
<employy>
    <name>
      张小三
      <sex>男  </sex>
      <birthday>1981 年 10 月 10 日 </birthday>
    </name>
      <name>
      王小妹
      <sex>女 </sex>
      <birthday>1985 年 12 月 12 日 </birthday>
    </name>
</employy>
```

12.4 本章小结

XML 与 HTML 一样，都是标准通用语言。XML 是 Internet 环境中跨平台的、依赖于相关内容的技术，是当前处理结构化文档信息的有力工具。本章介绍了 XML 的相关语法和术语，比较了 XML 和 HTML 之间的不同点，通过学习本章内容，能够很好地理解和掌握 XML 的使用方法。

12.5　思考与练习

1. 你认为 XML 究竟是什么技术?

2. XML 与 HTML 相比在商业应用中有哪些优势?

3. 什么是 XML 中元素的属性?

4. 试书写 XML 语句表示一个名字为 Student,仅仅具有 SNO 属性且该属性取值为“200”的空元素。

第13章 Visual Studio.NET 技术

万维网给全球信息的交流和传播带来了革命性的变化，为了更好地利用 Web 网络，人们开始探索利用网络来改变生活的方式，比如网上购物、收发邮件、远程教育等。

在网络开发平台上，微软的 ASP. NET 开发技术得到了广泛的应用。

通过本章学习，使读者明白，ASP 是如何过渡到 ASP. NET 中的，. NET 技术为何能构建 Web 应用程序功能。

学习目标及要求

（1）理解 Web 应用程序的主流开发技术。

（2）了解 Microsoft. NET 的战略。

（3）理解 Microsoft. NET 的开发平台构成。

（4）理解从 ASP 到 ASP. NET 的过渡过程。

13.1 .NET 架构

Microsoft. NET 开发平台包括一个用于加载和运行应用程序的新的软件基础结构（. NET Framework 和 ASP. NET）、一个新的开发环境（Visual Studio. NET），以及支持该结构的编程语言，如图 13-1 所示。

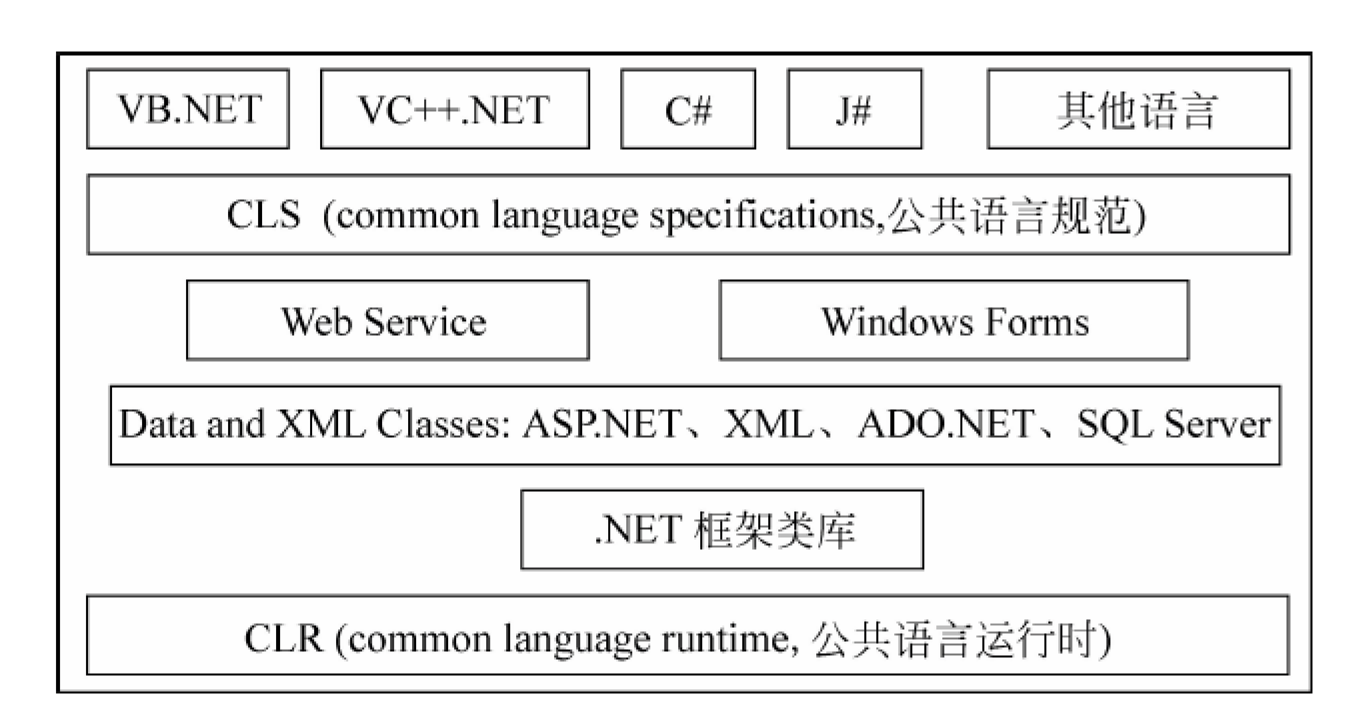


图 13-1 . NET 的体系结构

从图 13-1 可以看出，. NET 开发框架包括以下几个主要组成部分：公共语言运行时（CLR）；. NET 访问数据库的方法，如 ADO. NET 等；网络开发中的 Windows Forms 编

程；多种语言的支持，如 VB.NET、VC++.NET、C♯、J♯。

支持跨语言集成是.NET 的一大特色，跨语言集成又称为跨语言编程、混合编程或者跨语言互操作，为了达到这一目标，在.NET 框架中采用了一系列新技术。在.NET 框架中代码分为 managed code 和 unmanaged code 两类。前者称为托管代码或受控代码，后者称为非托管代码或非受控代码。在两类代码中，托管代码更体现了现代语言的风格，更完整地体现面向对象程序设计的思想。在本书以后的内容中，除非特别说明，我们的讨论都围绕托管代码展开。实际上，使用托管代码才能安全而又最大限度地利用.NET 的技术和资源。

托管代码能够在公共语言运行库中受到管理和服务，能够安全而高效地运行，为此，托管代码必须向 CLR 提供最低级别的信息——元数据(metadata)。全新设计的 C♯、Visual Basic .NET 和 J♯.NET 代码都受托管，而 Visual C++.NET 代码不受托管，但是编译器可以通过指定命令行开关 (/CLR) 来产生托管代码。

13.2 .NET 核心技术

.NET Framework 包括两个主要组件：

- 公共语言运行时 (CLR)。
- 统一的类库集。

13.2.1 公共语言运行时功能

CLR 和类库是.NET 框架的两个实体。CLR 的基本功能是运行代码并且提供各种服务，使得整个运行更加简易。CLR 的主要功能包括以下几方面：

(1) 执行和管理。完成运行过程中的代码加载、应用程序和托管对象的内存管理、对象自动布局、使用 JIT 将 MSIL 编译成本地代码并且控制代码执行、处理异常等全部步骤。

(2) 提供安全保证。类型安全，代码访问安全，控制托管和非托管代码的互操作。

(3) 对开发者的支持。提供界面信息、帮助调试等。

13.2.2 .NET 框架类库

.NET 类库(.NET framework class library)又称基础类库(base class library，BCL)，.NET 类库提供数千个类、接口等工具供程序员直接使用，主要包括数据结构、I/O 管理、数据库访问、多线程、远程访问、反射等。

.NET 支持的所有语言都能使用类库，任何语言使用类库的方式都是一样的，任何一种应用程序使用类库的方式也是一样的。

与 CLR 不一样的是：通常情况下，CLR 对程序员而言是透明的；而类库是程序员必用的工具，熟练掌握类库是程序员的基本功。

13.3 基于.NET的应用开发

在.NET应用开发中，主要使用如图13-2所示的应用开发环境。

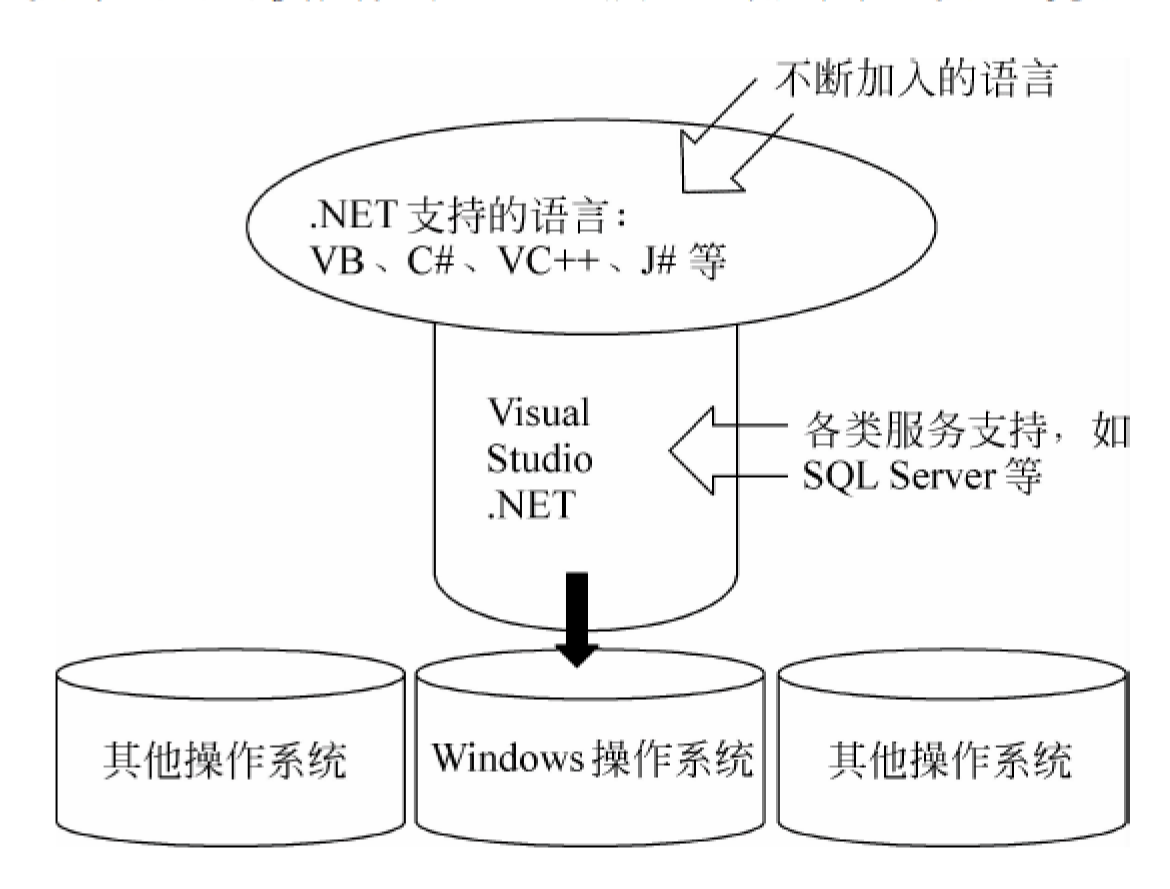


图13-2 .NET的应用开发环境的特点

1. 类库

在托管的环境里，开发扩展类库内容、自定义用户程序的类和组件是代码重用的有效方法，也是开发大型程序值得推荐的做法。.NET给出设计类库程序的规范，只有遵循这些规范，其他开发人员才能有效利用它们，使它们发挥最大的作用。

2. Windows应用程序

基于Windows Forms(Windows窗体)的应用程序是一种基于图形界面接口GUI的应用程序，一般是在用户计算机本地运行。

3. Windows控件库

创建Windows窗体应用程序的自定义控件。

4. 智能设备应用程序

Internet不仅仅连接PC机、服务器等计算机设备，还连接包括电视机、电冰箱、复印机等家用的、办公用的电器，可以通过Internet对这些设备进行远程控制。智能设备应用程序就是连接、控制这些设备，并且使它们互相通信的应用程序。

5. ASP.NET Web应用程序

ASP.NET是.NET中统一的Web开发平台，用来开发企业级的Web应用程序。ASP.NET是ASP的升级版，具有革命性的进步，程序员能够像设计Windows窗体应用一样设计Web应用；能够将代码与HTML文档分离，进行代码重用；能够在IDE中进行可

视化的开发。

6. ASP.NET Web Service

Web Service 是 Microsoft 的 .NET 战略中的重头戏，由于 XML 是当今 Internet 与各种系统之间的数据交换的标准，所以也称为 XML Web Services，它可以实现客户端-服务器或服务器-服务器间的数据交换。XML Web Services 采用松耦合的分布式结构，不与特定的组件技术或对象调用约定相关联，与编程语言、组件模型、操作系统等无关。一般的 Web 程序，例如浏览器，是人与程序的对话，而 Web Service 是程序与程序的对话。Web Service 是公认的 Internet 应用中最有影响的服务。

7. ASP.NET 移动 Web 应用程序

ASP.NET 移动 Web 应用程序是用于能无线上网的移动数字设备的应用程序，常见的移动设备有移动电话、掌上电脑等，.NET 提供移动控件来连接移动设备的设备适配器，在 .NET 上开发移动 Web 应用程序的过程与开发其他 Web 应用程序相似。

8. Web 控件库

在 Visual Studio.NET 环境下开发 Web 应用程序和开发 Windows 应用程序一样可以进行可视化的开发，其中最常用的图形元素是控件。和前面讲的 Windows 控件库应用程序类似，Web 控件库应用程序用来开发和扩展 Web 控件库，以及基于 Web 控件库的应用程序。

9. 控制台应用程序

控制台应用程序是一种简单而传统的程序形式，通常是字符型界面，可编译为独立的可执行文件，通过命令行运行，在字符界面上输入/输出。

10. Windows 服务

Windows 服务是服务器端的 Windows 应用程序，因为在服务器上运行，要响应客户端的请求，所以它可能被要求长时间运行，应该具有启动、暂停、重启动以及处理和其他程序关系的功能。

13.4 从 ASP 过渡到 ASP.NET

1996 年，Microsoft 在其 Web 服务器 IIS 3.0 中引入了 ASP 脚本技术，如人们熟悉的脚本 VBScript 和 JavaScript。借助于微软的工具，ASP 迅速成为在 Windows 下 Web 服务端的主流开发技术。

早期的 ASP 是 IIS 的一种开放式、无须编译的应用环境。ASP 程序是解释执行，在这个环境中，所有的程序在每次访问时，都需要 IIS 进行一次解释，客户端才会得到一个执行结果。

在 ASP.NET 中，所有的程序在执行前都是经过服务器编译的，这一点是 ASP.NET

和 ASP 的最大区别。ASP.NET 在程序执行的效率上也有很大的提高，许多功能是 ASP 无法达到的。

项目案例：ASP.NET 中的 Hello World 程序。

【题目】 Hello World 程序。

【代码】

```
//这是用 C# 编写的一个简单的 Hello World 程序
using System;                          //导入 System 命名空间
namespace Notepad                      //声明命名空间 Notepad
{
    class HelloWorld                   //声明 HelloWorld 类
    {
        public static void Main()      //程序入口点，Main 的返回类型为 void
        {
          Console.WriteLine("Hello World"); //控制台类的 WriteLine() 方法用于显示输出结果
        }
    }
}  //将文件保存为 HelloWorld.cs
```

【显示】

配置命令提示符以执行 .NET 控制台程序，如图 13-3 所示。

```
C:\WINNT\system32\cmd.exe

C:\Program Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\Bin>sdkvars

C:\Program Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\Bin>Set Path=C:\Prog
ram Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\Bin\;C:\WINNT\Microsoft.NET
\Framework\v1.1.4322\;C:\Program Files\Microsoft Visual Studio .NET 2003\Vc7\bin
\;C:\Program Files\Microsoft Visual Studio .NET 2003\Common7\IDE\;C:\WINNT\syste
m32;C:\WINNT;C:\WINNT\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Too
ls\BINN

C:\Program Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\Bin>Set LIB=C:\Progr
am Files\Microsoft Visual Studio .NET 2003\Vc7\lib\;"C:\Program Files\Microsoft
Visual Studio .NET 2003\SDK\v1.1\Lib\";C:\Program Files\Microsoft Visual Studio
.NET 2003\SDK\v1.1\Lib\

C:\Program Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\Bin>Set INCLUDE=C:\P
rogram Files\Microsoft Visual Studio .NET 2003\Vc7\include\;"C:\Program Files\Mi
crosoft Visual Studio .NET 2003\SDK\v1.1\include\";C:\Program Files\Microsoft Vi
sual Studio .NET 2003\SDK\v1.1\include\

C:\Program Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\Bin>Set NetSamplePat
h=C:\PROGRA~1\MICROS~1.NET\SDK\v1.1\
```

图 13-3 .NET 控制台的编译和执行

13.5 本章小结

.NET Framework 由 .NET Framework 类库和公共语言运行时两个主要组件组成。CLR 是管理用户代码执行的现代运行时环境，它提供 JIT 编译、内存管理、异常管理和调试等方面的服务。CTS 定义声明、定义和管理所有类型所遵循的规则，而无须考虑源语言，CLS 是所有针对 .NET 的编译器都必须支持的一组最低标准，以确保语言的互操作性。

13.6 思考与练习

1. 什么是 Web 服务器?
2. .NET 核心技术有哪些?
3. 公共语言运行时有哪些功能?
4. .NET 类库中有哪些主要内容?
5. ASP 和 ASP.NET 的主要区别是什么?
6. .NET 应用开发主要有哪些方面?

第14章 网站建设

如今越来越多的企业开始建立自己的网站，网站除了具有传播信息的功能外，还往往代表企业的形象。网站建设商业化和广告化的趋势，对网页设计的艺术性提出了更高的要求。虽然 HTML 语言提供了对常用 Windows 标准控件的支持，但是由于 Windows 标准控件的形状和颜色十分单一，网页的视觉效果可能会很差。因此有必要实现网页中的控件自定义、更加合理的结构布局、更加人性化的视觉效果、更加便捷的操作模式等，以满足网页设计艺术化、人性化的要求。于是，网页交互界面设计规划在实际项目中变得越来越重要了。

学习目标及要求

(1) 掌握网站的界面设计技巧。

(2) 熟悉常用网页版式设计。

(3) 能够独立运用色彩进行界面设计。

(4) 掌握网站的测试方法。

(5) 了解域名的种类和申请方法。

(6) 能够简单地推广自己的网站。

14.1 网站设计

网站设计包括网站使用的人机分析、界面的可用性设计、用户设计、色彩视觉设计、网页版式设计等内容。

14.1.1 界面设计

在网页界面设计方面，设计师应该考虑使用的便利性和视觉效果这两方面的问题。主要内容应该设计在引人注目的地方，所有操作性按钮位置应该科学，使用户容易操作；同时，要考虑符合网站主体的色彩，去除不必要的内容。

界面设计的要点。

- 醒目性：吸引用户的注意力，起到宣传的效果。
- 可读性：网站的内容通俗易懂。
- 鲜明性：主题准确、快速传达网站的构成内容。

- 造型性：维持整体外型上的稳定感和均衡性。
- 创造性：有鲜明个性和好的创意。

1. 导航

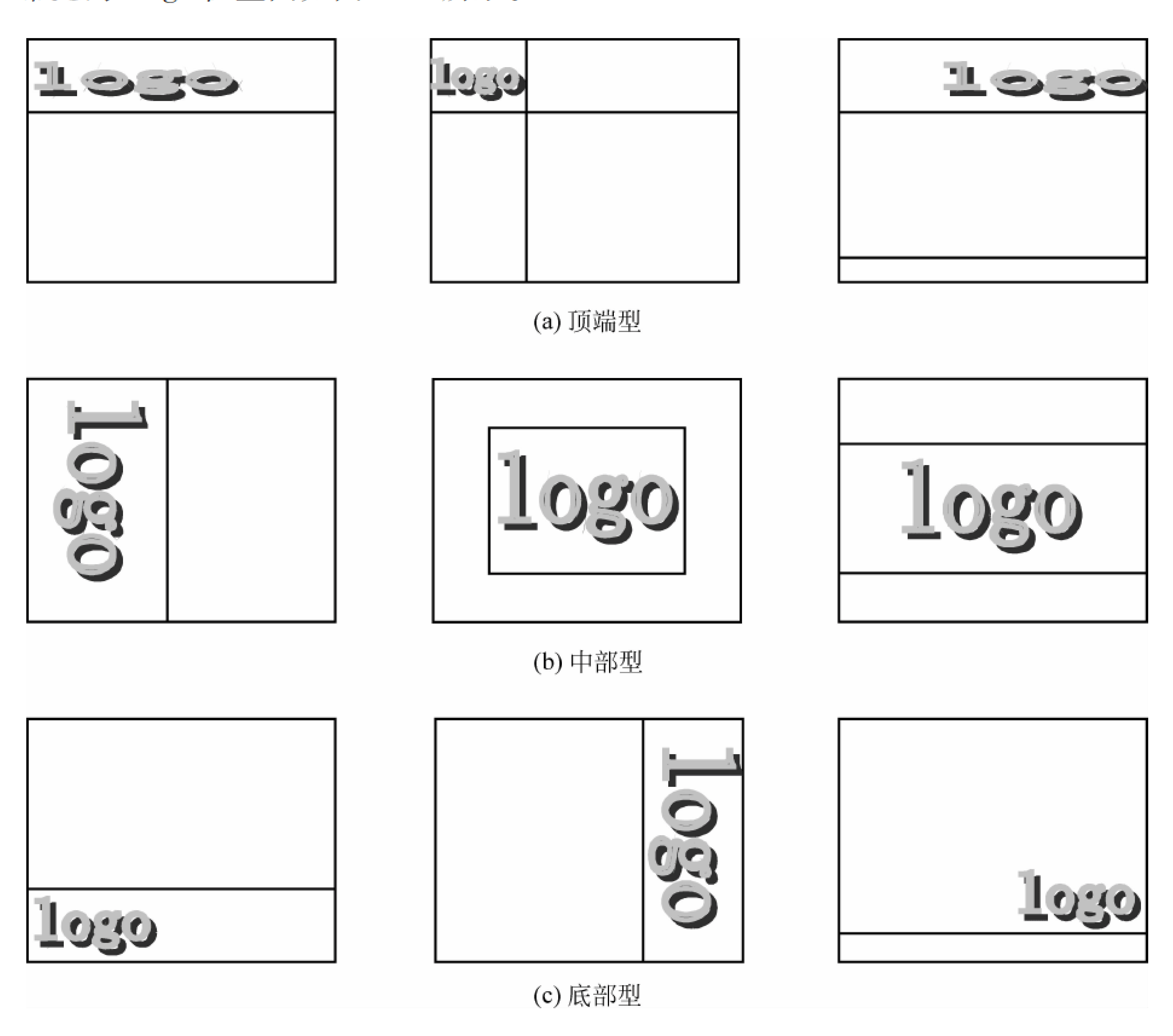
导航是最早出现在网页上的页面元素之一。导航就好像一本书的目录，先有章，而后是节，再就是小节。导航既是网站路标，又是分类名称，十分重要。导航栏应放置到明显的页面位置，让浏览者在第一时间内看到它并作出判断，确定要进入哪个栏目中去搜索他们所要的信息。

导航栏可分为横排导航、竖排导航、多排导航、图片式导航、Frame 框架快捷导航、Jump Menu 下拉菜单、隐藏式导航，甚至还可以是动态效果的 Flash 导航等。

2. Logo 标志

Logo 是标志、标识。Logo 的位置通常在页面的左上角，但它不是一成不变的。页面设计的个性化发展，需要设计师们的思考和定位，才能设计出个性的 Logo 标志。

常见的 Logo 位置图如图 14-1 所示。



(a) 顶端型

(b) 中部型

(c) 底部型

图 14-1　Logo 位置图

顶端型如图 14-2 所示；中部型如图 14-3 所示；底部型如图 14-4 所示。

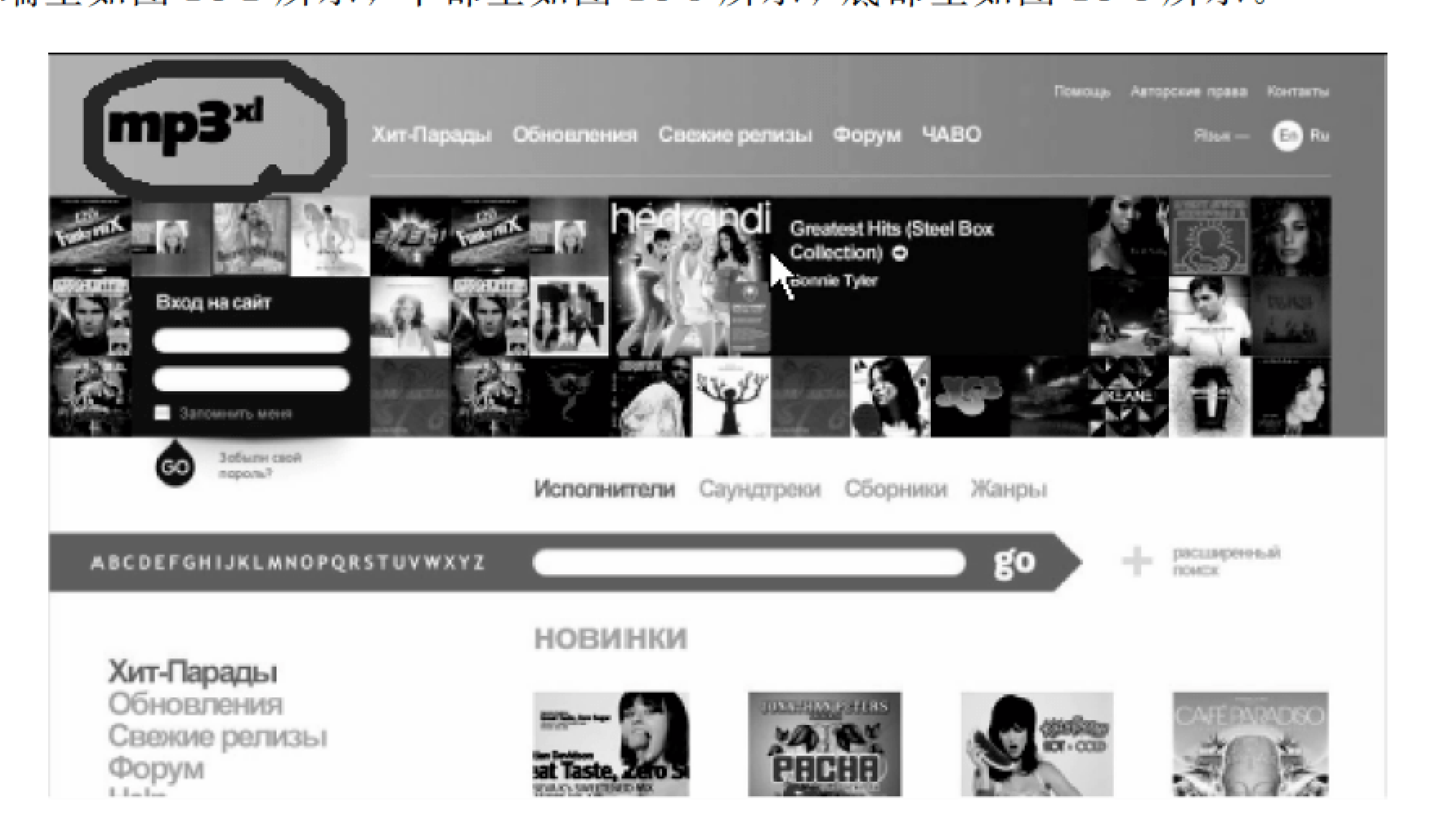


图 14-2 顶端型



图 14-3 中部型

3. 背景图

背景图的添加可以大大增加网站的整体效果，更加吸引人们的眼球。根据优秀网站的设计特点，总结出以下背景效果。

1）渐变色背景

某留学网站的官方站点如图 14-5 所示，页面设计清爽、大方。页面的主要效果来自网站背景图，该背景图是由蓝色渐变到白色，虽然没有蓝天白云，但是处处体现出蓝色天空的感觉，紧扣主题，色彩也十分鲜亮，使人感觉清爽，感官愉悦。

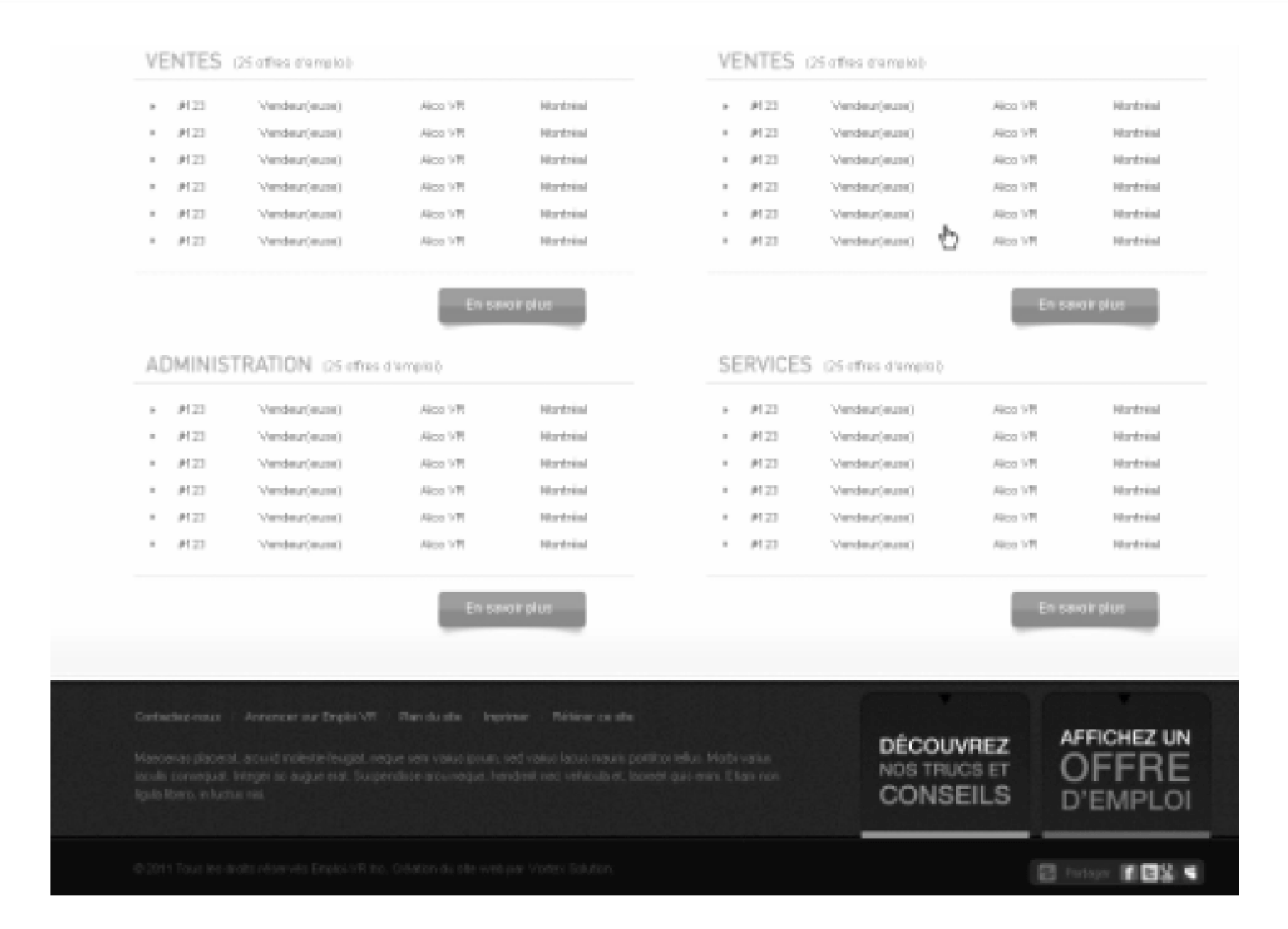


图 14-4　底部型



图 14-5　留学网站设计

2）图片背景

为了突出网页设计的个性化，很多网站甚至用整张图片作为背景，如图 14-6 所示。在使用整张图片做背景时，一方面一定要把下载速度考虑进去，网页不能承受大于 100KB 的背景图片；另一方面，不能因顾忌到下载速度而不断地压缩图片，否则图片模糊会影响页面效果。

为了不违反原设计意图，在保证图片清晰的情况下，可以从以下几方面从制作上优化网页：



图 14-6 汽车网站

- 把大的背景分割,不需要的部分省去,另一部分可以换成表格背景,分开插入页面当中。
- 注意到页面的长度,考虑好是否可以重复使用图片,以及是以 X、Y 哪个轴重复图片。
- 单色的地方可以使用表格背景色填充。有文字的地方可以考虑不做在图片里,这样压缩图片的时候就能选择.jpg 格式,不用担心文字压缩时会模糊,图片的字节数应该会小一些。
- 使用 CSS 来控制背景图在制作上会比较规范。

注意:若图片上含有文字,这些文字一定要清晰。模糊的文字会降低视觉效果。含有文字的图片尽量保存为.gif 格式和.png 格式,或者高质量的.jpg 格式。

4. 文字

互联网是一个很大的信息共享空间,而信息主要是以文字形式展现的。文字是页面的主要组成元素之一。针对不同需要,选择不同的字体、字号、内文编排和版式设计等,都属于文字的设计。

1) 字号的选择和信息文字的排版设计

信息文字的编排设计主要包括大标题、小标题、图片说明文、内文版式设计等,应注意以下几点:

(1) 每一行文字的长度最好为 20~30 个中文字(或 40~60 个英文字母)。

(2) 行距与字距已由软件内定。设计者可以利用段落与段落间空行及首行缩排方式来辅助阅读。

(3) 标题以 H1 到 H3 为佳,内文字体以 3~4 级为佳。

(4) 同版面字型最好在 3 种字型以内。

(5) 文字的颜色最好在 3 种颜色以内。文字在颜色上要与背景区分。

(6) 内文排列向左对齐并与左边界保持适当距离，可以利用表格填入文字以达到此效果。当然，根据诗文页面的美观要求，也可以向右对齐，网页设计可以更自由。

(7) 标题字数宜少，排在同一行，字数少则字可大，行可短，题意能在一瞬间即进入人们脑海。

(8) 内文中欲引人注意的重点，可用与内文同大小之斜体字或较粗的字体，或加下划线。利用 CSS 可以有多种方式注标强调点。

(9) 图片说明应靠近它所说明的图片，与该图合二为一。

(10) 可利用空行、画线条或装饰线分隔不同类型的内文。最好配合网页设计的风格来选择合适的装饰线，不要太过俗气，否则影响页面效果。

(11) 首字处理可为版面带来图案的趣味，同时不要太过夸张。

(12) 不同性质的内文，例如问题与解答，可分别采用不同字体或颜色。

2) 字体的选择

中文字体的运用也如英文字体一样，要考虑中文字体的个性情感问题。方正字库、微软字库、汉仪字库、经典字库都是很不错的。字体不宜太花哨，尽量不要使用太过个性的字体。如文鼎的一些字体，会显得轻浮和夸张，且不美观。也不要认为最常见的宋体没有魅力，在 Banner 中或在平面图片中运用粗宋体、老宋体等，编排起来都会非常和谐大方。

3) 字体的搭配

网上有很多中英文免费字体可以下载，设计者的计算机里通常也会装有上百种特殊字体。欲从繁多的字体中选择出几种搭配运用，使画面美观且阅读容易，需要注意以下几点原则：

(1) 大标题，小内文。利用字体大小的差异，表现标题及内文不同的重要程度。

(2) 粗标题，细内文。标题要粗，以最快的速度吸引人的注意。

(3) 字体少，字型少。同一页面内采用的字体宜在 3 种以内。

(4) 字体与网站内容相配合。着重理性类网站宜采用较冷静理智的方正型字体，如黑体、圆体；感性类网站不妨用较具变化感的字体。

4) 字族的运用

字族就是以某种字型为蓝本，将之变化为一组字体，它们有一个总称。例如，“黑体”字族有细黑、中黑、粗黑、特黑、超黑、反白立体、长黑、平黑、斜黑，但笔画形状都互相类似。页面可采用同一字族内多种不同字体来制造变化。字体间的相似性能让页面产生整个字体的印象，字体间的小差异也能使页面显得精致而有活力。

5) 字的变形设计

计算机制作网页图片时，可以在计算机上利用变形指令来改变字形，把原本正方形的文字变形成矮胖、瘦长或歪斜。将字拉扯挤压，做不规则的变形，这种变形字很难做到美观，因此一般设计上并不使用。

14.1.2　网页版式设计

界面交互主要集中在人机操作方面。一个成功的主页，单纯的美观是不够的，重要的是网页版式的设计。

布局的构成原则上是：统一、协调、流动、强调、均衡。

网页设计不是仅仅使用户在显示器上看到漂亮的装饰性画面那么简单。比起华丽的画面装饰，网页设计应该在更有效地传达信息方面多下些工夫。这必须有对布局设计和色彩的正确理解和相关知识，要熟知网页物理系统的特性，并具备应对各种情况的能力。在具备人性化界面的网站设计中，我们不难发现其表现出来的功能性和自然美。多数设计得比较好的网站都考虑了多媒体的特性，色彩设计和信息按照一定的秩序排列，能更有效地向用户传达内容。

根据目前网页设计现状，以及在网络上能呈现的设计条件，可以把网页框架分为三大类：分栏式结构、区域排版和非规律的页面排版。

分栏式结构和区域排版是两种较常见的基本构架形式，在商业网页设计方面有较广泛的应用，同时也是非商业网页设计构架的首选。

分栏式结构也可以称为竖分栏。这是一种超过一屏半，把页面从上到下分割为几列构架的设计结构，类似于新浪(sina. com. cn)的页面骨架设计。分栏式结构是竖长方形页面的框架。

区域编排利用辅助线、插图、色彩把网页平面分为几个规则的或不规则的区域。这种网页框架笼统称为区块分布式结构，也叫做区域编排式结构设计。横向割断分栏式结构，就可以变成区域排版，所以也可以称区域排版为分栏结构的变异。

区域编排的结构灵活、变化多，通常可利用颜色、线条、文字的断口或插图的变换等手段来划分区域。划分出来的区域的大小、形式都可以自行定义，格局则变化莫测。

此类框架适合于信息量小、信息源有规律的网站。区域编排不适用于门户开放式站点和信息交错紧密的站点，比较适用于企业站点和产品单一的电子商务类站点，以及对页面要求较高、信息流通量小的专业性信息站点。

以下列出几个较好的素材网站站点，供大家学习查阅参考。

- Kit(http://www. flashkit. com)：国际最著名的 flashkit，资源较多，教学、代码全面。
- ultrashock(http://www. ultrashock. com/)：提供 Flash 的资讯/高级教程和高级源文件，讨论区是最酷的 vbb，里面的版主都是知名的多媒体设计师。
- 韩国素材网(www. krwz. com)：专业的韩国素材网站，里面内容涵盖网站、界面设计等内容。
- 素材中国(www. sccnn. com/)：提供平面素材、网页素材的网站。
- 闪客帝国(http://www. flashempire. com)：业内新闻、动态排行、技术支持等。
- 闪客天空(http://www. flashsky. com)：源代码非常多，便于下载学习。

14.1.3 色彩设计

网页色彩是设计色彩中的一种。无论是单个页面的色彩搭配，还是整个网站的色彩设计，首先要确定主体色(主色调)。主体色可以是一种，也可以是几种。通常我们形容一个页面的颜色时会这样说：榕树下(http://www. rongshuxia. com)是绿色的；腾讯(http://www. qq. com)是蓝色的；新浪(http://www. sina. com. cn)是黄色的；而雅虎(http://www. yahoo. com)是白色的。

例如，某商务网站如图 14-7 所示，其主体色为黑色，整个页面是由由浅及深的黑色组成的。该网站的主要卖点是照相机，网站整体色彩采用照相机的颜色——黑色，配上人物照片，凸显出动感、时尚。网站下端为白色，整个画面虚实结合，空间感极好。整个页面色彩感觉典雅、古色古味，既展示了相机的魅力，又体现出动感的元素，与照相机配合起来十分贴切。



图 14-7 照相机网站

1. 色彩的三要素

明度、色相、彩度是构成色彩关系的 3 个最基本的要素。了解色彩的基本概念和规律，有助于我们进行实用色彩设计。

色相即每种色彩的相貌、名称。如红、橘红、翠绿、湖蓝、群青等。色相是区分色彩的主要依据。

明度即色彩的明暗差别，即深浅差别。色彩的明度差别包括两个方面：一是指某一色相的深浅变化，如粉红、大红、深红，都是红，但一种比一种深；二是指不同色相间存在的明

度差别，如六标准色中黄色最浅，紫色最深，橙色和绿色、红色和蓝色处于相近的明度之间。

纯度即各色彩中包含的单种标准色成分的多少。纯的颜色色感强，所以纯度亦是色彩感觉强弱的标志。不同色相所能达到的纯度是不同的，其中，红色纯度最高，绿色纯度相对低。

阳光通过三棱镜的折射，可分解成红、橙、黄、绿、青、蓝、紫 7 种颜色。在这 7 种光谱色中，红、黄、青是 3 个最基本的色彩，称为"三原色"或叫"第一次色"。用这 3 种色彩调配而成的色彩称为"间色"或叫"二次色"，由二次色调配而成的色彩称为"复色"或叫"三次色"。由三原色、间色、复色组成一个过渡自然的色相环是基本的色相，如图 14-8 所示。

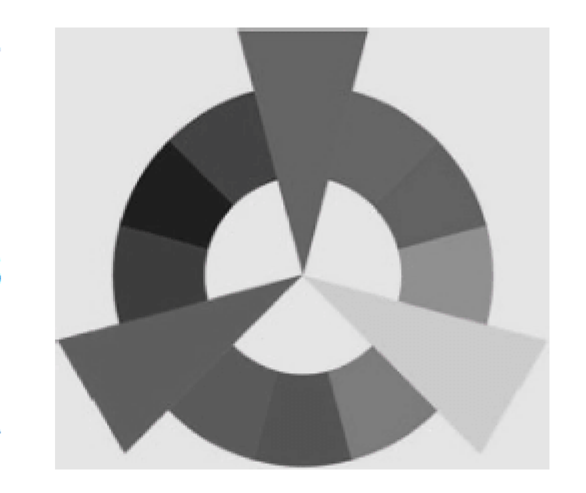


图 14-8　三原色色环

明度的变化往往通过区分其色阶来表示，从白到黑，一般分为 9 级或 13 阶。明度分成 3 组，可表示为高明度、中间明度和低明度。各个色彩的彩度各不相同，明度也不同。高纯度的色彩加白、加黑或加其他低纯度的灰色，都会降低它们的彩度。黑、白、灰只有明度的差别，没有彩度的变化。在配色中常常将这 3 种无彩色作为调和、区隔的因素来使用。

2．色彩对比关系

色彩对比强弱取决于色相环上的距离（角度），距离（角度）越小对比越弱，反之则对比越强，如图 14-9 所示。

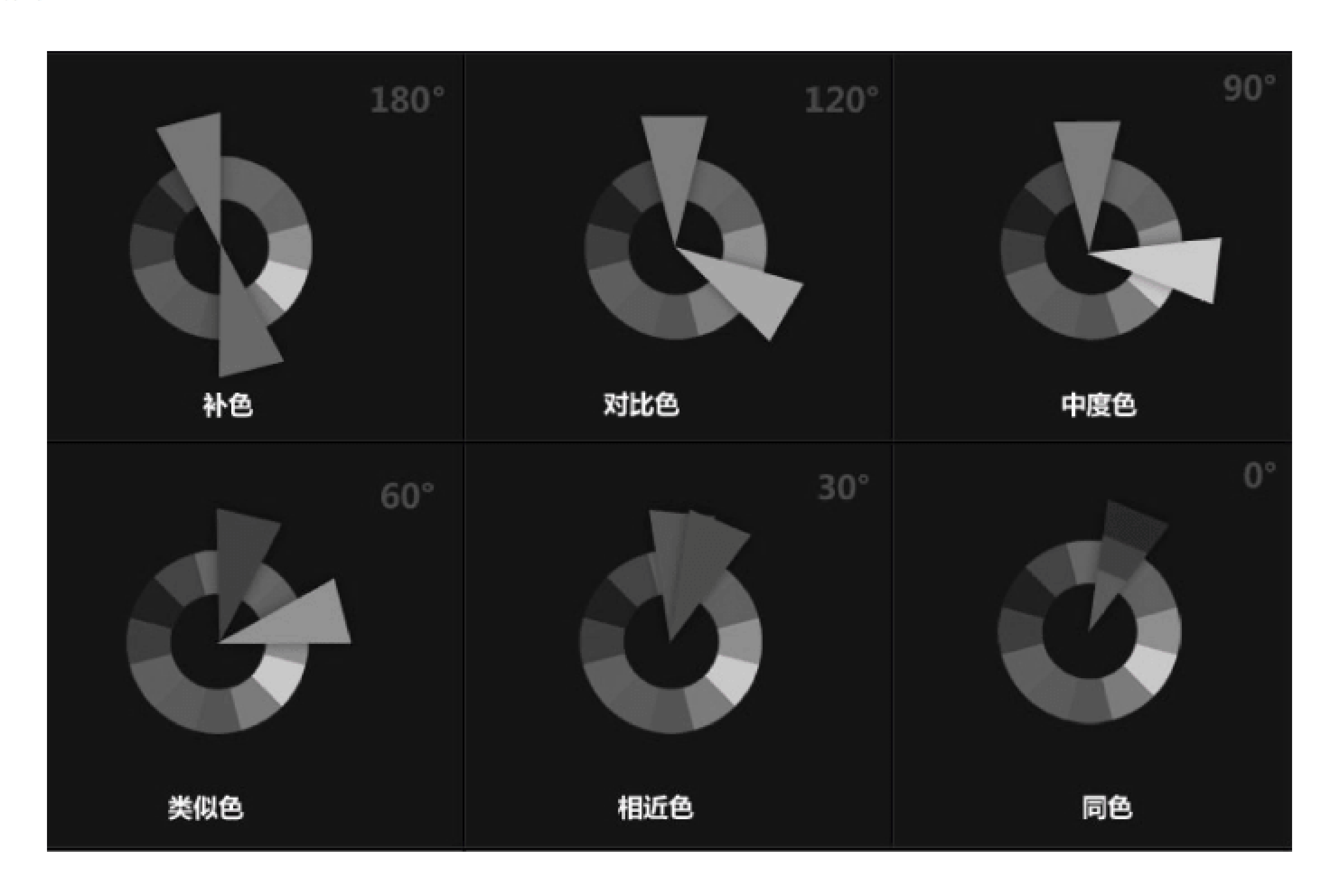


图 14-9　色环的强弱对比

1）明度对比

明度对比是所有色彩对比的基础，是决定画面情调、作品风格的重要因素。要区分无彩色和同种色比较容易，而要区别不同色彩之间的明度差要困难些。在以明度为基准配色时，明度对比强则视敏度高，画面清晰明朗；明度对比弱则视敏度低，画面形象模糊不清，或显得柔和暧昧；明度对比太强时则使色彩层次两极分化，带有强烈的刺激性，有时也会显得单调。

2）色相对比

因为色相差异而产生的对比称为色相对比。在色相环中任何两种以上色相的配置都会产生对比作用。对比的强弱与色彩的相对位置和距离有关，在色轮中邻近色相的对比较弱，对比最强的是 180°补色对比。例如红与绿的对比，白与黑的对比。

3）彩度（纯度）对比

因为彩度差异而产生的对比称为彩度（纯度）对比。彩度是色相的彩度，人们的眼睛在感觉不同色相时所能区分的彩度是不同的。高彩度（纯度）的色相明确、鲜艳、纯净，低彩度（纯度）的色相含蓄、柔和。

4）面积对比

色彩的面积对比是指各色相间在搭配时的大小、主宾对比关系。对同面积的黄色和紫色来说，其色彩是有很大差异的，黄色显轻，有前进感；而紫色显重，有后退感。在配色时，为了求得页面的主色调，常用面积大的色彩或色彩组合来起作用，起到统帅主调的效果，这种以色彩为主的配色就是面积对比的方法。

5）补色对比

补色对比是一种特殊的色相对比。在色轮中任何一种原色（红、黄、蓝）与其另外两个原色之间色的对比称为补色对比。补色对比是最强烈的对比，在设计中采用这种对比的配色能鲜明地突出传达的主题和个性，视敏度很强。

6）冷暖对比

冷暖对比是指两种不同色性系列在配色中的对比关系，以红色代表暖色系列，以蓝色代表冷色系列，但在实际的配色中这种冷暖对比的关系有时是明确的，有时只是同种色在彩度上的差异或同类色在色彩倾向性上的微妙差异。把握冷暖对比的分寸对准确传达信息主题有不可忽视的重要性。

【案例分析】

（1）美体饮食网站。

美体饮食网站如图 14-10 所示，采用柔和的色彩体现出美食与美体之间的关系。



图 14-10　美体饮食网站

美体饮食网站的色相对比关系分析如图 14-11 所示。



图 14-11　美体饮食网站色相对比关系

（2）音乐盛典网站。

音乐盛典网站如图 14-12 所示，用明亮的蓝色表达出动感的音乐，人物背景对比鲜明。



图 14-12　音乐盛典网站

音乐盛典网站的色相对比关系分析如图 14-13 所示。

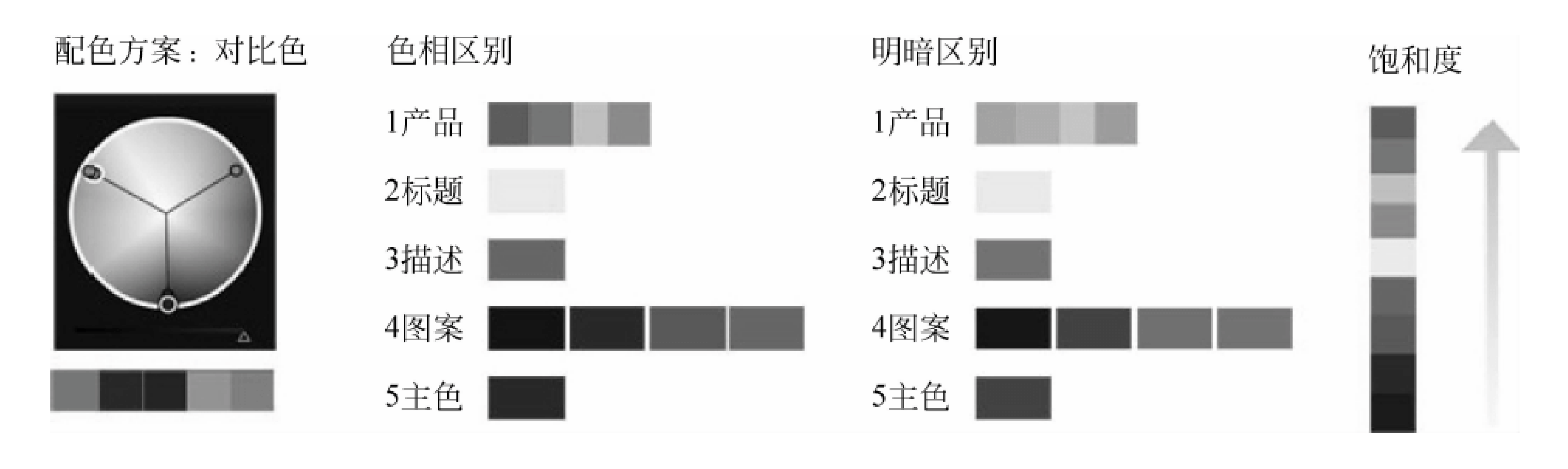


图 14-13　音乐盛典网站色相对比关系

（3）游戏网站。

游戏网站如图 14-14 所示，采用墨绿色和黑色体现出神秘、魔幻的感觉，符合网站的特点。

游戏网站的色相对比关系分析如图 14-15 所示。



图 14-14　游戏网站

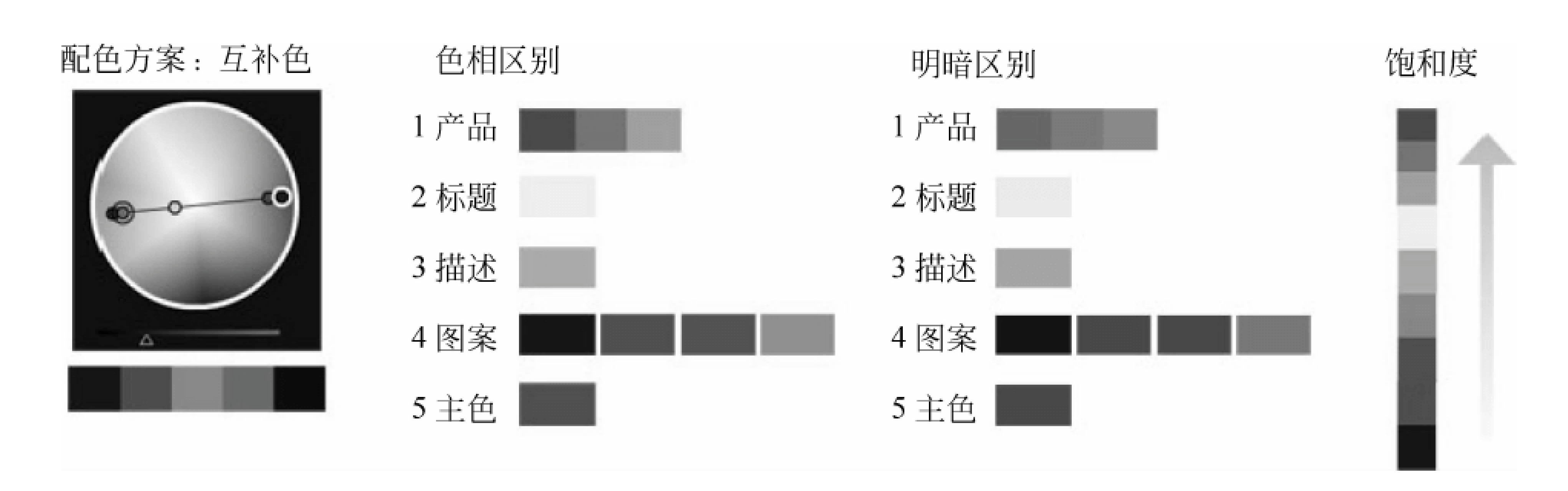


图 14-15　游戏网站色相对比分析

3. 色彩的心理感受

色彩的心理感受是指视觉对色彩的反应，随外在环境而改变。视觉受色彩的明度及彩度的影响，会产生冷暖、轻重、远近、胀缩、动静等不同感受与联想。色彩就本质而言，并无感情，而是人们在生活中积累的普遍经验的作用，形成人们对色彩的心理感受。

1）红色

具体联想：火焰、太阳、血、红旗、辣椒。

抽象情感：热烈、青春、朝气、积极、革命、活力、健康、温暖、愤怒。

2）橙色

具体联想：橘子、柿子、秋叶。

抽象情感：快活、温情、健康、欢喜、和谐、任性、疑惑。

3）黄色

具体联想：灯光、闪电、黄金、金发、香蕉、菠萝、柠檬、蛋黄、枯叶、银杏树的落叶、稻穗、黄沙。

抽象情感：轻快、明快、朝气、希望、快乐、富贵、轻薄、未成熟、刺激。

4）绿色

具体联想：大地、草原、庄稼、森林、蔬菜、青山。

抽象情感：自然、健康、成长、新鲜、安静、和平、凉爽、清新。

5）蓝色

具体联想：天空、海洋、水。

抽象情感：沉静、平静、科技、理智、速度、年轻、青年、寂寞、冷淡、冥想、阴郁、诚实、真实、可信。

6）紫色

具体联想：葡萄、萝卜。

抽象情感：优雅、高贵、细腻、神秘、不安定。

7）黑色

具体联想：夜晚、黑夜、黑发、黑烟、乌鸦、煤炭、阴暗。

抽象情感：沉着、厚重、老年、古典、不吉利、悲哀、绝望、恐怖、死亡、地狱、心地阴险。

8）白色

具体联想：雪、云、雾、白纸、白布、天鹅。

抽象情感：纯洁、清白、纯粹、纯真、清净、明快、和平、轻薄、空白。

9）灰色

具体联想：水泥、鼠。

抽象情感：平凡、谦和、失意、中庸。

常用色彩语言的一般意义如下：

(1) 红色是最引人注目的色彩，具有强烈的感染力。它是火的色、血的色，象征热情、喜庆、幸福，又象征警觉、危险。红色色感刺激强烈，在色彩配合中常起着主色和重要的色调和对比用途，是使用得最多的颜色。

(2) 黄色是阳光的色彩，象征光明、希望、高贵、愉快。浅黄色表示柔弱，灰黄色表示病态。黄色在纯色中明度最高，与红色色系的颜色配合产生辉煌华丽、热烈喜庆的效果，与蓝色色系的颜色配合产生淡雅宁静、柔和清爽的效果。

(3) 蓝色是天空的色彩，象征和平、安静、纯洁、理智，又有消极、冷淡、保守等意味。蓝色与红、黄等颜色运用得当，能构成和谐的对比调和关系。蓝色的网页在网络上到处可见，它是大家竞相使用的主要颜色之一。为了能够在众多蓝色的站点中脱颖而出，配色要巧妙，设计者应选择不同的蓝色，尝试新的搭配。

(4) 绿色是植物的色彩，象征着平静与安全，带灰褐的绿色则象征着衰老和终止。绿色和蓝色配合显得柔和宁静，和黄色配合显得明快清新。由于绿色的视认性不高，多作为陪衬的中性色彩运用。如果用绿色做网页的主色调，注意应该选择墨绿或其他稳重的绿色，搭配黄色等用以调和，否则会感觉轻浮和不自然。

(5) 橙色是秋天收获的颜色，鲜艳的橙色比红色更为温暖、华美，是所有色彩中最暖的色彩。橙色象征快乐、健康、勇敢。

(6) 紫色象征优美、高贵、尊严，另一方面又有孤独、神秘等意味。淡紫色有高雅和魔力的感觉，深紫色则有沉重、庄严的感觉。紫色与红色配合显得华丽和谐，与蓝色配合显得华贵低沉，与绿色配合显得热情成熟。这些色彩运用得当能产生新颖别致的效果。

(7) 黑色是暗色，是明度最低的非彩色，象征力量，有时又意味着不吉祥和罪恶。黑色和许多色彩构成良好的对比调和关系，运用范围很广。

(8) 白色是表示纯粹与洁白的颜色,象征纯洁、朴素、高雅等。作为非彩色的极色,白色与黑色一样,与所有的色彩构成明快的对比调和关系,与黑色相配,则构成简洁、明确、朴素、有力的效果,有很好的视觉传达能力。

色彩的冷暖感觉一般是由色相决定的,如火一样的红、橙、黄等色,会使人产生温暖感,称之为暖色;像水一样的蓝色、蓝绿色或倾向于蓝、蓝绿色称为冷色;绿和紫一般看做中性色,但如果它们倾向于蓝,就可归于冷色系列,倾向于红、黄,就可归于暖色系列。一般来说,暖色系列的色彩较易使人感觉温暖,冷色系列的色彩较易使人觉得寒冷。

红、橙、黄等暖色系列的纯色给人以兴奋感,易表现快乐的情绪。蓝、蓝绿等冷色系的纯色给人以沉静感,常常使人联想到和平、安静、休息和生机盎然。它还是一种使人镇静的颜色。黑白对比强烈的色彩给人以紧张感,灰色及彩度低的色彩给人和谐而舒适的感受。

政府、一部分企业站点应该选择沉静一些的配色,而娱乐、体育等站点则应该倾向于兴奋一些的配色。

4. 色彩设计方法与技巧

色彩设计的一般程序如下:

(1) 整理和收集资料。要使色彩设计有较强的目的性和针对性,力求避免许多习惯的、用之过多的配色手法,应充分发挥色彩的信息传达功能,(产品类、企业类网站)可广泛收集与商品、竞争商品、消费者生活习惯、文化传统、心理期望等有关的资料和网站,并根据商品的市场目标情况进行分类整理,以获得必要的信息。

(2) 分析审视宣传主题,选择合理的色彩表现方法。如何把握你所要表达的主题,以色彩的语言传达和强化这一主题,需经过认真的审视和推敲。在这一过程中色彩本身不只是被动地陈述,完全能够利用其独特的语言优势进行再创造、再提高,以强化网站的主题和风格,使受众留下长久的、难以遗忘的深刻印象。对主题的内容和精神的分析和理解,也不能建立在经验的基础上,应该以创新为基点。

一般来说,网站主题的特点是可以类型化的,可分为以说理为主的或以感情诉求为主的;可分为以传统文化为背景的或以时代风尚为背景的;也可分为以消费观念为前提的或以生活习性为前提的;还可分为以消费对象为主体的或以商品功能、特色和意义为主体的多种选择。网站宣传的主题一经确认,色彩表现的方向就基本明确了。

对于个人主页情感类的网站,就更加需要强化这一环节。色彩独特的感染力会左右和主导网站的整体风格。

(3) 选择合理的色彩配置方法。色彩配置的方法有多种,在确认表现主题后,可根据实际情况进行选择。可以依据主题表现的重点进行配色,如理性的或感性的色彩搭配、色调和风格的选择;是强调商品色还是强调情绪色;是强调一贯色还是强调流行趋势。可以色相为主进行配色,也可以色彩的明度或彩度为主进行配色。

(4) 调整色彩的细微关系。调整阶段的主要任务是以总体特征为依据,进一步推敲色彩的各种关系,如色调、色相、明度、彩度、色性、面积比、空间关系、位置关系等,使色彩的配置能准确、完整地传达信息,烘托气氛。以"商品色"为依据的配色即所谓的"商品色"是指某些或某类商品在消费者的心目中已经形成的、固定不变的色彩形象。在长期的广告宣传作用下,某些商品的包装、色彩等外观形态已经与商品的内在特性融为一体,成为不可分离的

基本特征。以色彩调子为基础的配色，色调的选择和控制对色彩设计有举足轻重的作用。色调是指整体的色彩关系，如以色相来区分，有红调子、黄调子、蓝调子、绿调子等；如以明度来区分，有高调子、中间调和低调子；如以彩度来区分，有鲜明调子和灰调子；如以色彩关系来区分，有对比调子和调和调子，等等。把握好色彩的基本调子，就能使色彩的配制协调统一、形象突出。

注意：在线配色器的网址：

（1）http://kuler.adobe.com/。

（2）http://www.jd100.net/ip/color.htm。

（3）http://colorschemedesigner.com/previous/colorscheme2/index-en.html。

项目案例1：页面中的色彩配置。

【题目】 在网页设计实践中，如何进行页面内部的色彩分配呢？可以以相对标准的分栏式为可视模板，用以比较不同的配色方案，以及产生不同的视觉感受。

【效果】 参考模型如图14-16所示。

1区：网页抬头（上部标志区、Banner区）。

5区：网页标题。

3区：网页内容（文字图片区）。

2区：侧栏（功能区、附加信息区）。

4区：页脚（版权位置、其他元素信息）。

图14-17是由橘红的暖色系变化到褐色的中性色系，选择的都是单一型的配色，如绿色系、蓝色系、玫瑰红。在图14-17和图14-18中，可以看到在这一小型的缩略图中，整个页面的色调所发生的根本性变化。有时候这样做效果不错，但却并不是经常能奏效。

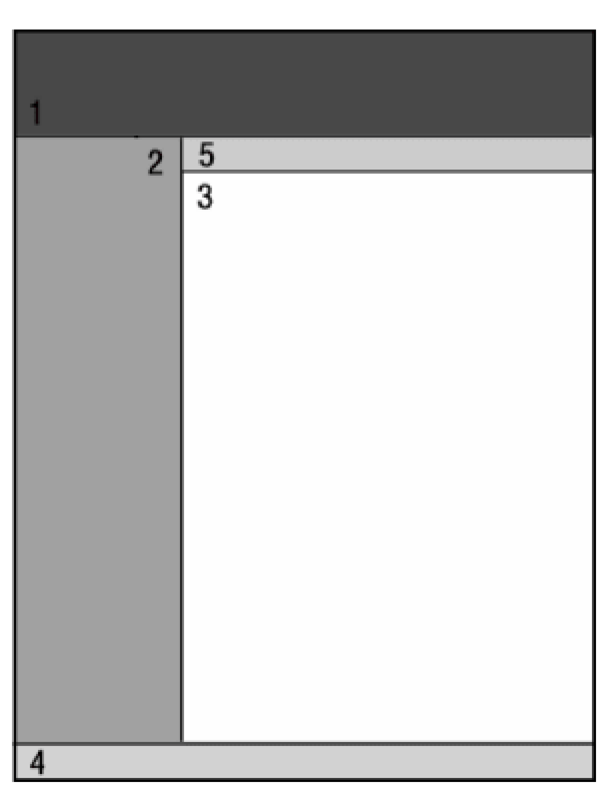
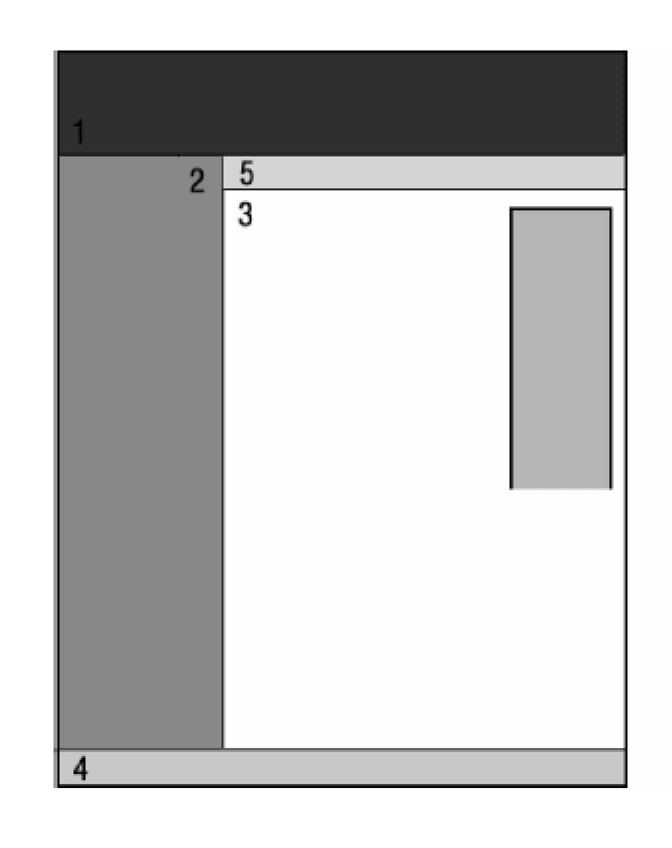
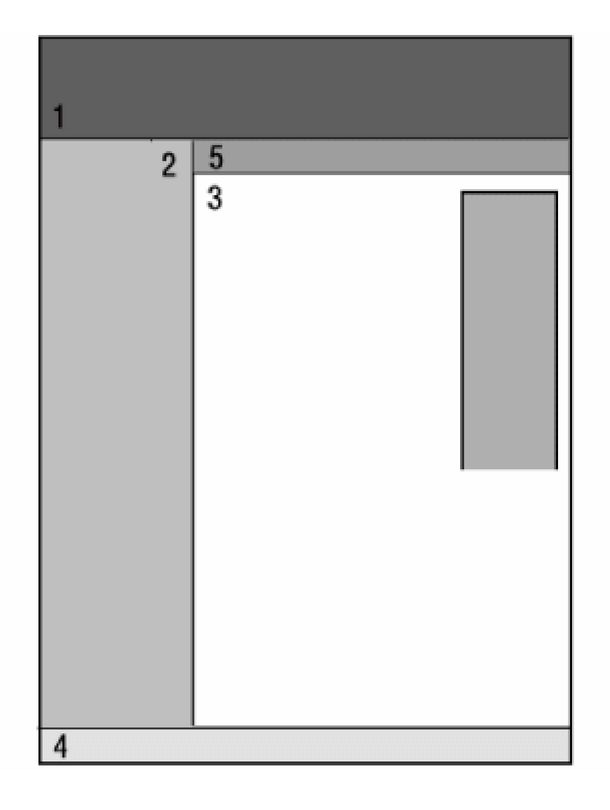


图14-16 参考模型





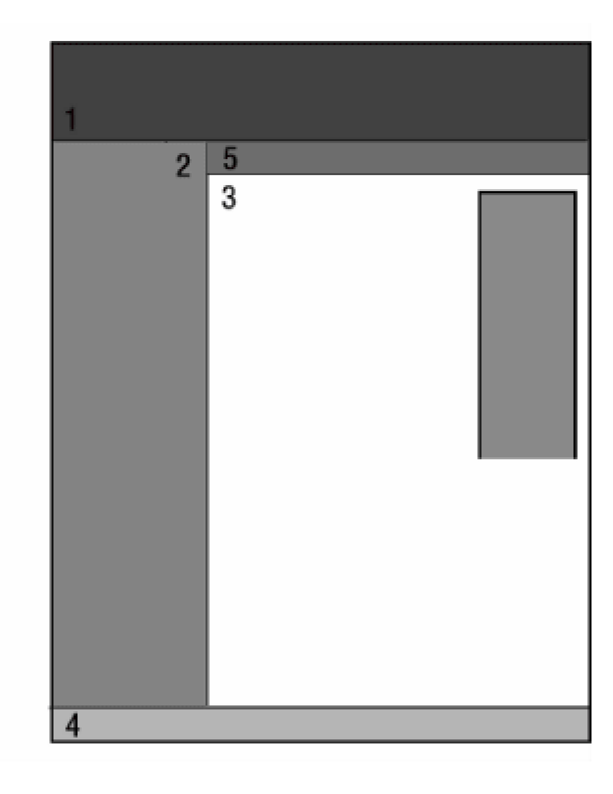
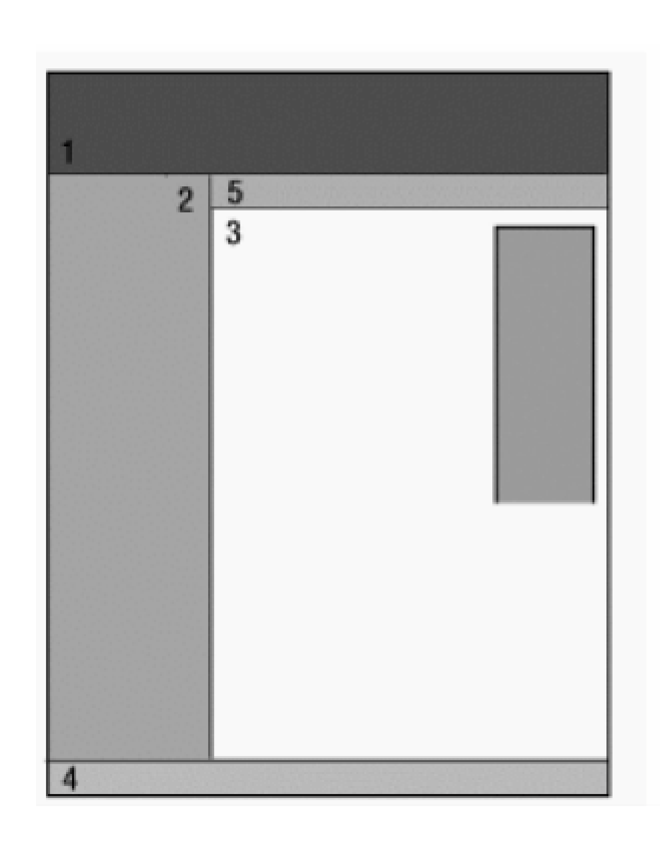
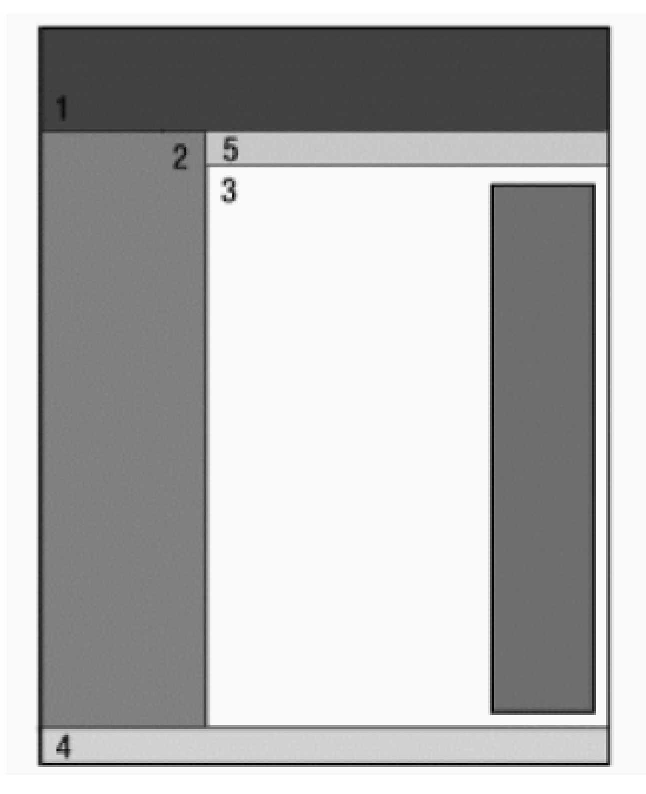


图14-17 模型设计（一）

如果无法把握单一颜色的数值，可以从Photoshop自带的色彩方案中取色，如图14-19所示。

多数设计师认为画出美妙绝伦的页面才是设计，其实不然，要在有限的元素和色彩里表达出风格才是更有难度也更有层次的设计。经过严谨的思考和漫长的推敲，最终只花几分





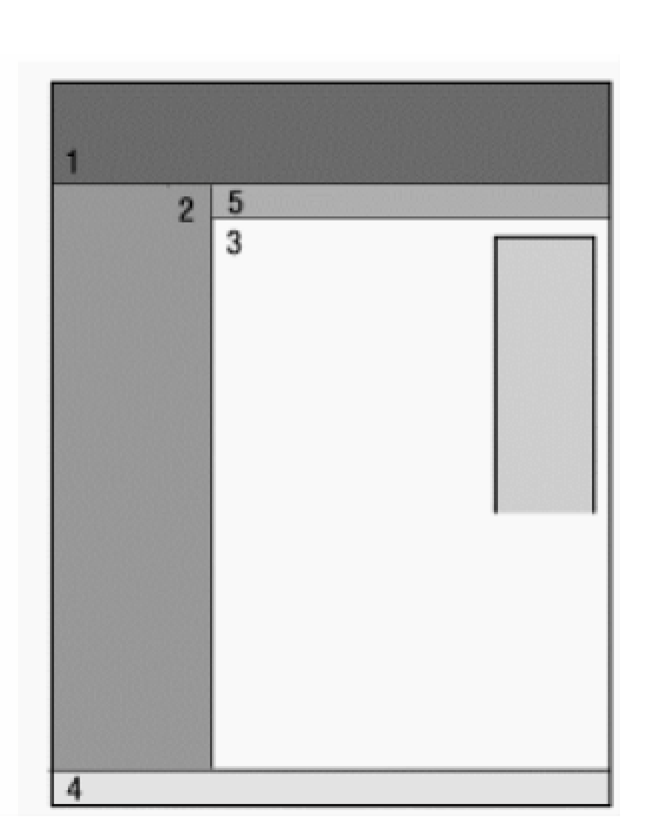
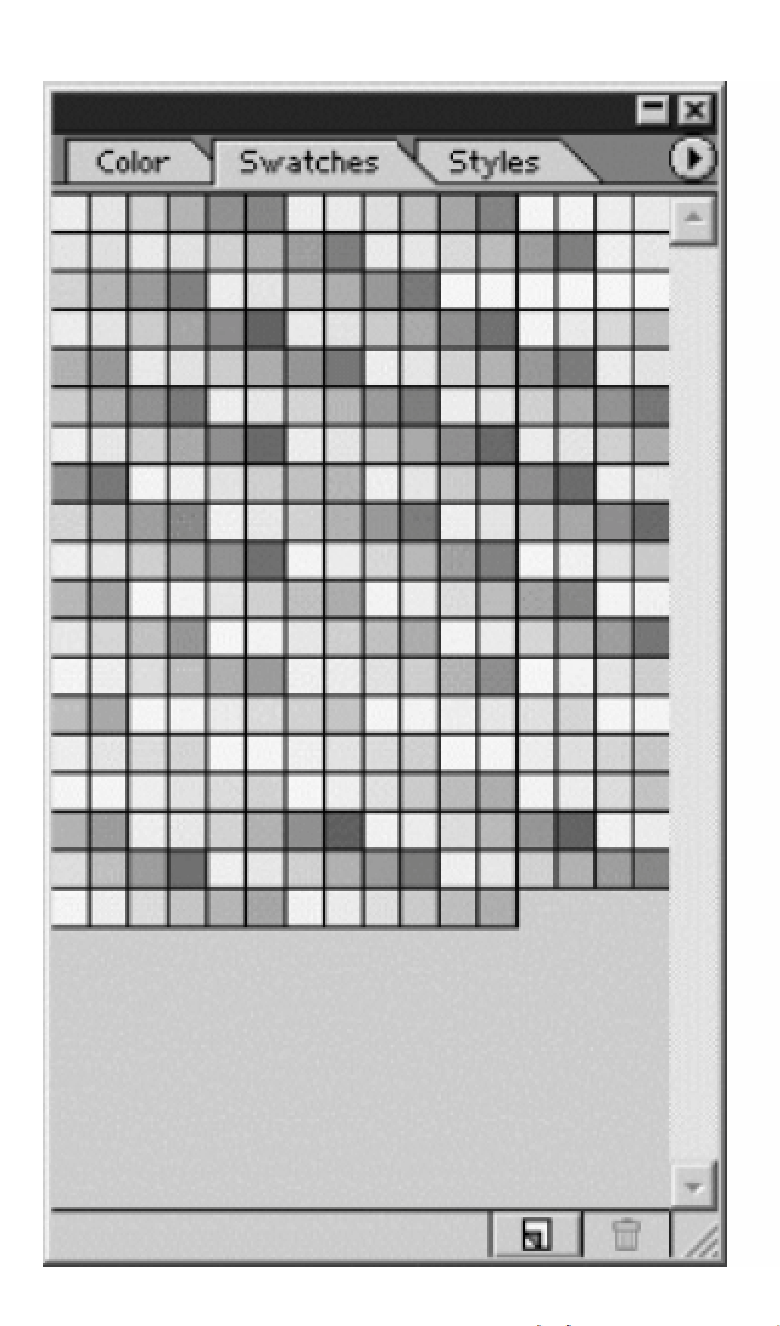


图 14-18 模型设计(二)



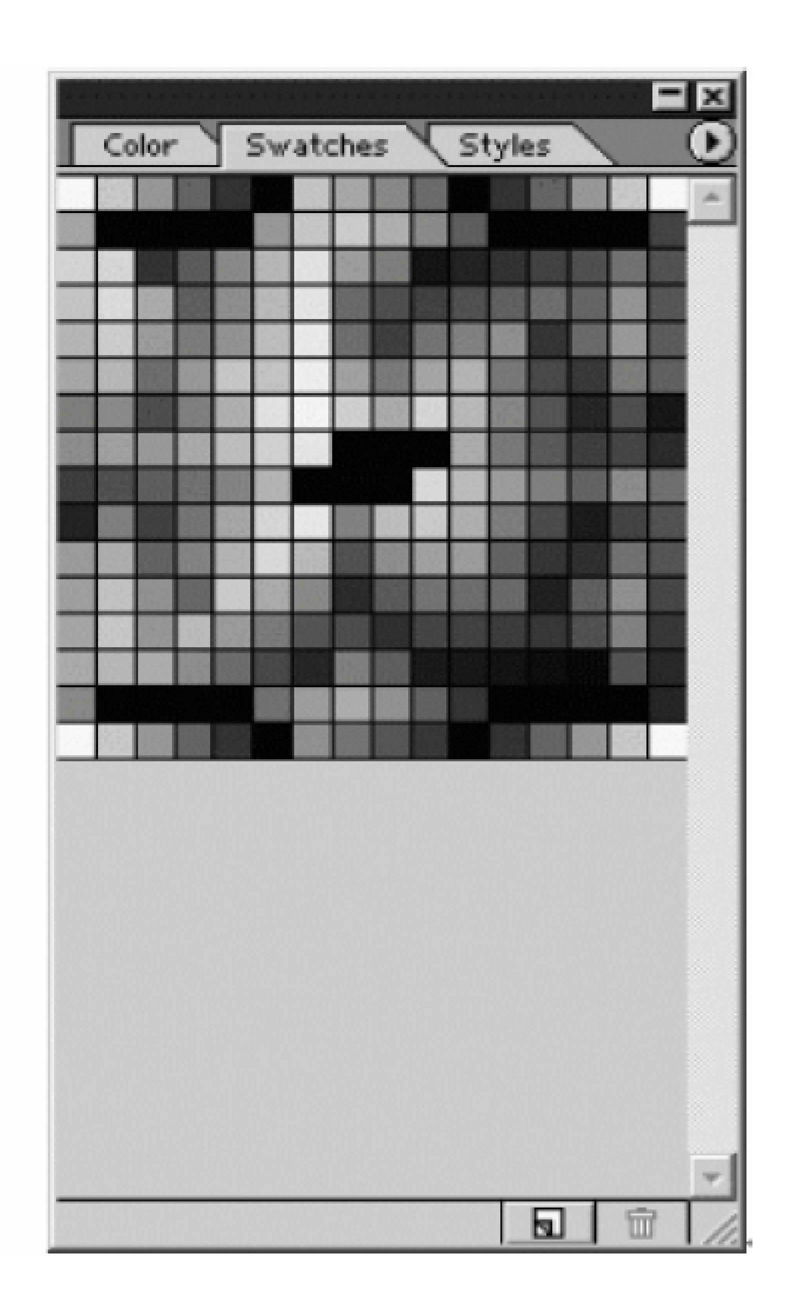


图 14-19 Photoshop 色板

钟为页面画下一个简单的圆角框或者一条两像素的实线,其效果也许比涂满了色彩和组合了数十个分层的画面更好。设计最终将上升到思想的层面,而不是仅仅停留在工具和素材的使用。

14.2 网页制作过程

网页制作过程如下:

(1) 与客户沟通需求,直到需求确定。

(2) 根据需求里的栏目设置和设计要求等,开始设计页面,直到确定。

(3) 对设计出来的页面进行制作,如果有开发的话,可以编写程序。

(4) 制作页面的步骤(用 Photoshop 或其他的作图软件进行切图,用 Dreamweaver 建立站点,新建页面并开始制作首页,然后制作栏目页,再制作内容页,最后把相互之间的链接做好)。

(5) 上传到服务器进行测试。

(6) 正式上线。

14.3 测试网站

网站制作好之后,要将网站上传到虚拟主机空间里。这里主要介绍有关上传软件 LeapFTP 的使用方法。

操作步骤如下:

(1) 下载并安装 LeapFTP 的最新版本。

(2) 运行 LeapFTP,如图 14-20 所示,选择 1 处("站点"→"站点管理器"),弹出"站点管理器"窗口;在 2 处选择 My Servers 选项,然后在 3 处单击"添加站点"按钮,在弹出的窗口 4 处输入你喜欢的站点名称(如 mysite)。



图 14-20　LeapFTP

(3) 单击"确定"按钮后,出现如图 14-21 所示对话框,在 1 处输入你的站点地址(如 abc.com),取消对 2 处"匿名"选项的选中,在 3 处输入 FTP 用户名(如 abc),在 4 处输入 FTP 密码(取消对"保存密码"选项的选中),单击 5 处的"应用"按钮。

(4) 切换到"高级"标签(6 处),如图 14-22 所示,在 1 处选中刚建立的站点名称(mysite),在 2 处选择"使用安全"选项,在 3 处的"传送模式"处选择"二进制",单击 4 处的"应用"按钮。至此,你的 FTP 站点已经构建成功。

(5) 可以单击"连接"按钮,直接进入连接状态;也可以回到主菜单单击"FTP 服务器"→My Servers→mysite 进入连接状态,如图 14-23 所示。



图 14-21　站点管理器

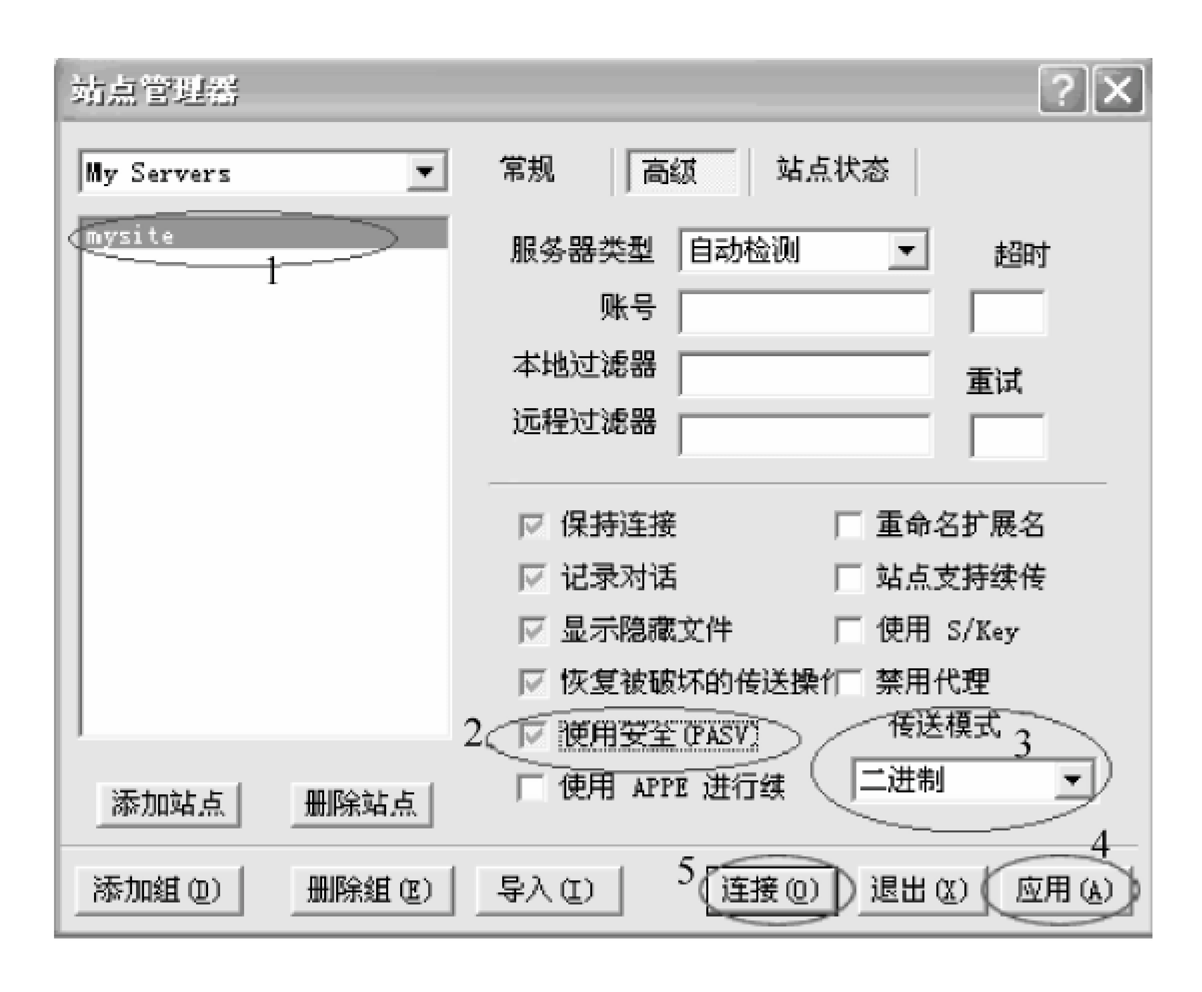


图 14-22　高级标签窗口

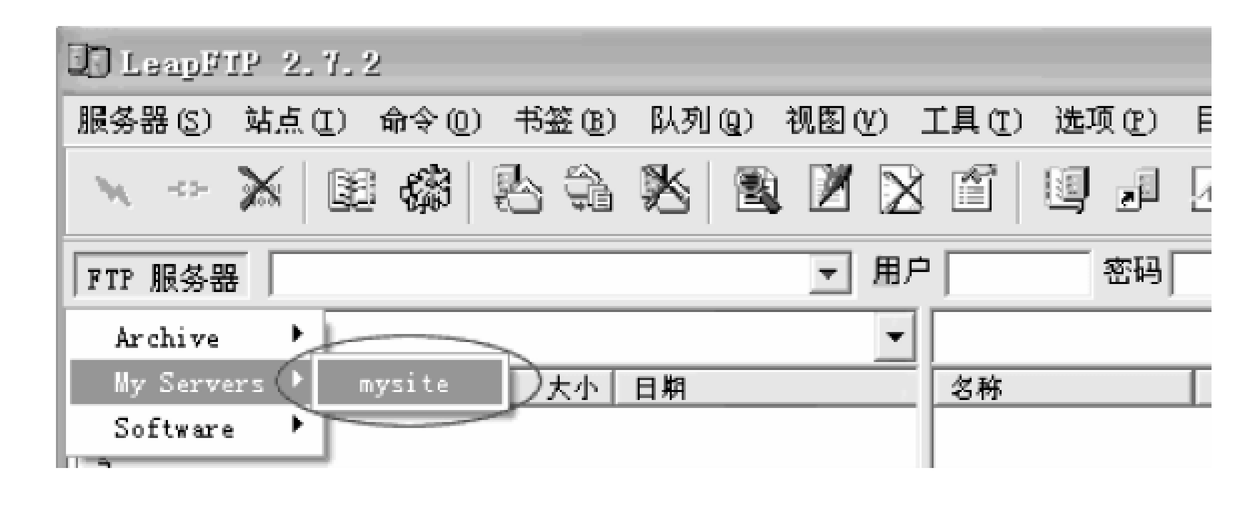


图 14-23　连接按钮

（6）连接成功后的界面如图 14-24 所示。左上框为本地目录，可以通过 1 处的下拉列表框选择要上传文件的目录，选择后该目录下的所有文件都会显示在这个框里。在要上传的文件上右击，选择“上传”命令，这时在队列栏里会显示正在上传及未上传的文件，当文件上传成功后，在命令栏里会出现“传送完成”的提示，此时在过程目录栏里就可以看到上传的文件了。

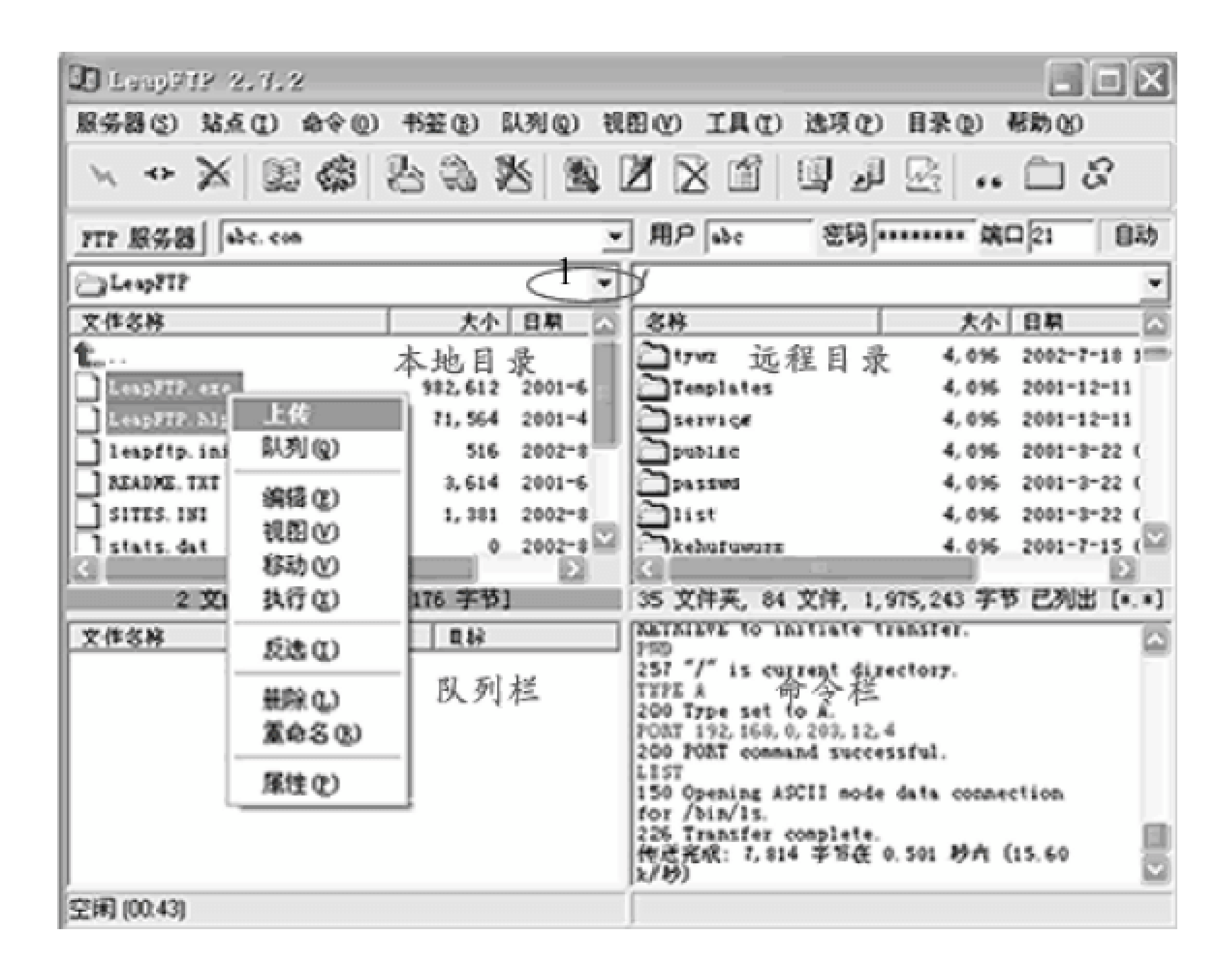


图 14-24 连接成功

14.4 域名介绍

域名是在网站开发后期必须要申请的一个名称，这个名称不仅代表着网站的寓意，同时也体现了网站的内涵。如果域名得不到正确解析，人们看不到它，即使网站做得再好也没有用。所以说，域名和网站名称都是十分重要的。

一个具有吸引力的域名应该具有在刹那间吸引人的能力。比如 cool. com，曾经看过或听过该域名的人都被这个极具吸引力的“cool”所吸引，人们都好奇地想看看 cool. com 是个怎样的网站，并有打开这个网站的冲动。这样的域名就非常成功。同理，有人注册了 Chinadesigner. org，还没有设置任何内容时，每天都会有 1500 次的点击率，就是这个原因。

当别人听过或看过一次便留下深刻印象的域名，其价值是显而易见的。设计“简洁”非常重要，因为只有简单的域名才便于记忆和有利于宣传推广，相信没有人会喜欢记住那些复杂的域名。即使你建了一个很好的网站，做了大量的广告推广，但域名过于复杂难记，结果也可能会徒劳无功。

虽然有好多域名很容易记，但如果同你所开展的商业活动没有任何关系，用户就不能将你的域名同你的商业活动联系起来。现在国内网站的域名层出不穷，可见中国人的创意真的发挥得淋漓尽致了。许多国内网站的域名设计思路都不错，以下是一些值得参考的域名设计技巧以及设计域名应注意的几方面：

(1) 简单清晰。一个域名在人们看了第一眼后能够记住的程度有多少，是否给人留下深刻而良好的印象，简单清晰是关键因素。从视觉方面考虑，尽量避免多个连续相同或类同的字母，如 nhumv、CGGCO，这种域名让人看得眼花缭乱。从发音方面考虑，尽量使用单音节或双音节的组合。如 bar、nosame、qiye 等。

(2) 含有良好的寓意。域名的意义是指域名本身的含义，域名的意义表达越清晰，效果就越好。iloveu——“我爱你”，直接表达爱意。book4sale——开门见山，向人表示出售书籍。findtoys——站在儿童的角度，想找玩具吗？来这里。mycar——给人亲切的感觉！有

人说国内数字域名潮是丁磊注册163后掀起的，不管怎么样，数字式域名正在渐入民心，相信大部分网民随便都可以说出一大堆。利用中国人独有的拼音做域名，这样的域名读起来感觉非常亲切，甚至不用刻意去记。由数字加英文组成的域名，例如51job，利用“51”的谐音译成“无忧”，这种搭配可谓独具匠心。

（3）留意商业机会。域名设计思路最好围绕你所进行的商业活动来考虑，带有行业性质、类型的更佳。

（4）给域名减肥。当你已经设想好了域名，但长度太长，你可以对其进行“减肥”。

1. 什么是域名

域名是Internet上的一个服务器或一个网络系统的名字，在全世界，没有重复的域名。域名的形式是以若干个英文字母或数字组成，由“.”分隔成几部分，如cndns.com就是一个域名。

2. 域名与网址有什么区别

一个完整网址范例如下：http://yourdomain.com。对应于这个网站的域名则是yourdomain.com。人们建立一个提供WWW信息的主机后，以域名来为其命名。此时，这台主机的名字称为www.域名。当访问者要访问这台主机时，浏览器会以指定的HTTP(hypertext transfer protocol)协议向主机发出数据请求。为此，我们描述一个完整的网址时都会加上前缀http://。

3. 中国企业可以注册的域名有哪些

互联网上的域名可谓千姿百态，但从域名的结构来划分，总体上可把域名分成两类：一类称为“国际顶级域名”（简称“国际域名”）；一类称为“国内域名”。

4. 什么是国际顶级域名

一般国际域名的最后一个后缀是一些诸如.com、.net、.gov、.edu的“国际通用域”，这些不同的后缀分别代表了不同的机构性质。

- ac：科研机构。
- com：工、商、金融等商业性的机构或公司。
- edu：教育机构。
- gov：政府部门。
- net：互联网络、接入网络的信息中心(NIC)和运行中心(NOC)或公司。
- org：各种非盈利性的团体或组织。

5. 什么是国内域名

国内域名的后缀通常要包括“国际通用域”和“国家域”两部分，而且要以“国家域”作为最后一个后缀。以ISO31660为规范，各个国家都有自己固定的国家域，如cn代表中国，us代表美国，uk代表英国等。

可以按照你的机构性质选择适合你单位使用的三级域名。行政区划域名是按照中国的

各个行政区划分而成的，其划分标准依照原国家技术监督局发布的国家标准而定，包括行政区域名 34 个，适用于我国的各省、自治区、直辖市。例如，北京的机构可以选择如 cnnic. bj. cn 的域名；www. abcd. cn 就是一个中国国内域名；www. cndns. com 就是一个国际顶级域名。

6. 为什么要注册.cn 域名

. cn 域名是由我国管理的国家域名(ccTLD)，属于顶级域名。当前选择 .cn 域名注册的企业越来越多，超越 .com，在全球具有最大的市场。按照行政省份划分，有 .bj.cn、.sc.cn、.sh.cn 等。

注册 .cn 域名有如下优点：

(1) 显著的标识作用，体现自身的价值和定位。

(2) .cn 域名资源丰富，可以获得满意的域名。

(3) 适用中国法律，全面保障用户利益。

(4) 全中文服务，保障用户知情权。

(5) 注册手续简便、快捷。

(6) .cn 域名使用更有保障。

(7) 有利于维护国家互联网信息安全。

7. 出现域名争议怎么办

任何机构或个人认为他人已注册的域名与个人的合法权益发生冲突的，均可以向争议解决机构提出投诉。可以与 CNNIC 认证的域名争议解决机构联系争议事宜。目前的争议解决机构有两家：中国国际经济贸易仲裁委员会域名争议解决中心和香港国际仲裁中心。

8. 域名注册完成后，要注意些什么

(1) 按时续费。注册人需要每年向注册服务机构交纳域名运行管理费用。按管理办法的规定：年域名续费截止日同申请日。对于续费截止日未完成续费的域名，将暂停服务。所有域名暂停服务 30 天内仍可正常续费。

(2) 如果注册信息发生变化，应当及时通知域名注册服务机构加以变更。

(3) 注意保存注册服务机构提供给你的用于更改信息的密码和转移注册服务机构的密码。

.com、.net、.org 是国际最广泛流行的通用域名格式，这些域名任何人都可注册。

14.5 网站维护与推广

14.5.1 网站维护

网站维护是指网站运营后需要做的维护工作。维护工作包括后台服务器杀毒、补丁更新、垃圾清理、删除过期专题栏目、界面美化、网站内容更新、广告更新等。与其他媒体一样，网站也是一个媒体，需要经常性地更新维护才会起到既定的商业效果。网站一旦上线运营，网站的管理与维护就摆在了企业的面前。

1. 网站的软硬件维护

计算机软硬件在使用中常会出现一些问题。比如，网络设备中硬盘、主板出现问题，将导致企业网站不能正常运行。由于非专业人员的误操作，导致整个企业网站瘫痪。没有任何操作系统是绝对安全的。维护操作系统的安全必须留意操作系统网站的更新补丁，及时为本系统安装升级包或者打上补丁。如 WindowsXP、SQL Server 等服务器软件都要及时打上补丁。服务器配置本身就是安全防护的重要环节。有不少黑客利用了没有正确配置的微软 IIS 服务产生的漏洞攻击网站，使网站造成了不可估量的损失。

2. 网站内容更新

建站容易维护难。对于网站来说，只有不断地更新内容，才能保证网站的生命力，否则网站不仅不能起到应有的作用，反而会对企业自身形象造成不良影响。

对于网站内容更新，网站管理人员无须懂得任何的网页制作技术，只要了解基本的计算机文本处理方法，就能够方便地在网站上更新或维护信息内容。

一般情况下，网站内容更新需要做以下几个工作：

(1) 添加、修改及删除信息。

(2) 按不同栏目对信息分类，方便信息的管理及查找。

(3) 信息可按标题、发布日期、关键字等不同分类进行查询。

(4) 网站维护人员当天更新的内容，及时发布到网站前台。

(5) 对广告内容要做及时更新。

【备注】网站管理员在网站维护中起到了决定性的作用。要成为优秀的网站管理员，需要具备以下技能：

(1) 熟练使用 HTML 语言制作静态网页。

(2) 熟悉 Photoshop 和 Flash 基础，能够制作和设计网页界面。

(3) 了解 JavaScript，熟悉 JavaScript 的作用和使用方法。

(4) 对 CSS 深入了解。必须会使用 CSS 制作网页，并注重不同浏览器的兼容性。

(5) 熟练使用 Dreamweaver。

(6) 熟悉数据库的操作。熟练掌握 SQL Server、Access 的操作方法。

(7) 熟练掌握 SQL 语句的使用。

14.5.2 网站推广

这里主要介绍免费的推广方法。

1. 参加各种广告交换组织

可以到诸如 http://www.webunion.com 或者 http://www.linkexchange.com 这样的网站去登记，成为它们的会员，把它的广告加到你的主页上，而你的主页图标也会出现在其他会员的主页上。

2. 与相关网站做友情链接

目前绝大多数网站都有宣传的积极性,因此大多数站点都愿意与别人的主页做友情链接,也就是设有“互换链接”栏目。

3. 利用电子邮件群发送消息

如果你加入了一个电子邮件群,就可以在你的站点上开辟一个小的栏目来专门介绍该电子邮件群所讨论的内容,然后向该电子邮件群地址发一封信,告诉他们在你的网站中有专门栏目介绍这样的内容,请他们给予更多指导。邮件群的成员收到这封信后,一般都会乐意到该网站去访问,利用这种手段也可以达到推广站点的目的。

4. 通过新闻媒体进行宣传

如果你的网站确实很有特色或者你的网站能够提供一些能获取利润的投资建议,给访问者带来一定的利益,可以借助电视、广播等媒体进行一系列报道,或者可以写一些有关自己站点特色内容的报道寄到比较有影响的报刊、杂志社,寻求他们的帮助。

5. 利用留言板进行宣传

如果你的网站有类似公告栏或留言板的功能,完全可以在这些位置放上你网站的地址,一旦其他网友在浏览你的留言或公告栏时,可能看到你网站的地址而顺便去拜访你的主页。

6. 到各大报刊发表文章

作为一名计算机爱好者,学习和使用计算机的时间肯定不会太短,而且在理论和实践中也肯定积累了不少经验和心得,这时最好不要把这些经验独享,应该用语言把它们表达出来,与其他同行或网友进行真挚交流。

7. 利用电子邮件发出通知

如果你手中有许多朋友或者用户的电子邮件地址,可以考虑利用电子邮件通知他们来访问。利用这种方法来宣传主页的关键之处在于,要留心收集用户电子邮件的地址,拥有的电子邮件数量越多,就意味着你的主页蕴藏着更大的访问量。即使是这样,要注意你的邮件内容应包含一些有用的东西,尽量不要发那种垃圾邮件。

8. 在聊天室里发出邀请

很多人上网的一个主要目的就是交流,想交到更多的朋友。交友当然首先就去聊天室了。在聊天室里你可以适时地向这些网友发出邀请,请他们访问你的主页,并谦虚地请他们给你的网站建设提出一些宝贵建议,同时在与他们聊天之际,可以把你的站点所具有特色的内容大肆宣传一番,以引起网友们足够的兴趣和注意。

9. 在新闻组上发布你的主页

参加过网络新闻组的网友们可能经常在相关的新闻组讨论区中见到诸如“某网站今天

正式开张啦！”、“某主页最近进行了更新，增添了许多特色的内容”、“欢迎各位网友常到某网站做客！”等信息，其实就是这些网站的版主在通过新闻组来宣传自己的主页。

10. 使用专门注册工具提交到搜索引擎中

目前搜索引擎已经有好几千个了，如果一个一个地申请并将自己的网页也加入进去，这样可能要花费几个月的时间。此时可以使用专门的主页注册软件，它只需几分钟的时间就能自动在搜索引擎的相关目录下和主题下注册。

11. 加入 links 类站点或申请测评

网络上 links 数据库类型的站点越来越多，如丹麦酷站联盟（http://www.cwd.dk）、中国的蚁盟（http://www.yimeng.org）等。这类站点通常有很多类别，但大多以艺术设计为主。如果你的站点设计得好，就可以申请加入。能否加入要视每个站点的评委要求，有一些站点的要求非常高，如英国的 linkdup（http://www.linkdup.com），而另一些则相对低一些。

有经济条件的网站在推广中，可以利用提供免费服务来起到宣传作用，如免费信箱、免费空间、免费留言本、免费计数器、免费 BBS、免费调查。

14.6 开展电子商务

电子商务（electronic commerce）以电子及电子技术为手段，以商务为核心，把原来传统的销售、购物渠道移到互联网上来，打破国家与地区有形无形的壁垒，使生产企业达到全球化、网络化、无形化、个性化、一体化。有关电子商务的通俗定义是：电子商务是指利用简单、快捷、低成本的电子通信方式，买卖双方不谋面地进行的各种商业和贸易活动。

电子商务是以计算机网络为基础。以电子化方式为手段，以商务活动为主体，在法律许可范围内所进行的商务活动过程。

电子商务的范围很广，一般可分为企业对企业（business-to-business，B2B）或企业对消费者（business-to-consumer，B2C）两种。另外还有消费者对消费者（consumer-to-consumer，C2C）这种大步增长的模式。随着国内 Internet 使用人数的增加，利用 Internet 进行网络购物并以银行卡付款的消费方式已渐流行，市场份额也在迅速增长，电子商务网站也层出不穷。电子商务最常见的安全机制有 SSL（安全套接层协议）及 SET（安全电子交易协议）两种。

电子商务是一个不断发展的概念，电子商务的先驱 IBM 公司于 1996 年提出了 Electronic Commerce（E-Commerce）的概念，到了 1997 年，该公司又提出了 Electronic Business（E-Business）的概念。但我国在引进这些概念的时候都翻译成电子商务，很多人对这两者的概念产生了混淆。事实上，这两个概念及内容是有区别的，E-Commerce 应翻译成电子商业，有人将 E-Commerce 称为狭义的电子商务，将 E-Business 称为广义的电子商务。E-Commerce 是指实现整个贸易过程中各阶段贸易活动的电子化。E-Business 是利用网络

实现所有商务活动业务流程的电子化。E-Commerce 集中于电子交易,强调企业与外部的交易与合作,而 E-Business 则把涵盖范围扩大了很多,广义上指使用各种电子工具从事商务或活动;狭义上指利用 Internet 从事商务或活动。

以下介绍几种常用的电子商务模式。

1. B2B(Business to Business)

B2B 指的是商家(泛指企业)对商家的电子商务,即企业与企业之间通过互联网进行产品、服务及信息的交换。通俗的说法是指进行电子商务交易的供需双方都是商家(或企业、公司),他们使用了 Internet 的技术或各种商务网络平台,完成商务交易的过程。这些过程包括发布供求信息,订货及确认订货,支付过程及票据的签发、传送和接收,确定配送方案并监控配送过程等。B2B 有时写做 B to B,但为了简便干脆用其谐音 B2B(2 即 to)。B2B 的典型是百万网、阿里巴巴、百纳网、中国网库、中国制造网、商机网、敦煌网、慧聪网、瀛商网、中国 114 黄页网、太平洋门户网等。

2. B2C(Business to Customer)

B2C 指的是商家对消费者的电子商务模式。B2C 模式是我国最早产生的电子商务模式,以 8848 网上商城正式运营为标志。B2C 即企业通过互联网为消费者提供一个新型的购物环境——网上商店(卓越亚马逊、天天团购网、新蛋网、京东商城、红孩子商城、团火网、当当网等),消费者通过网络在网上购物、在网上支付。

第一种:综合商城

代表:淘宝、天河城、贝斯尼尔、正佳广场

综合商城就如我们平时进入的大型商场一样,一楼是金银饰品,二楼是女士服饰,三楼是男士服饰,四楼是运动/装饰,五楼是手机数码,六楼特价……将 N 个品牌专卖店装进去,这就是商城。淘宝商城就是这种形式。淘宝商城有庞大的购物群体、稳定的网站平台、完备的支付体系和诚信安全体系,既促使了卖家进驻卖东西,又促使买家进去买东西。如同传统商城一样,淘宝自己是不卖东西的,但提供了完备的销售配套。

网上商城在人气足够、产品丰富、物流便捷的情况下,其低廉的成本、二十四小时的不夜城、无区域限制、更丰富的产品等优势,使其获得交易市场的一个重要角色。

第二种:专一整合型

代表:赛 V 网、李宁网

赛 V 网主要从事体育用品网上销售、导购、新闻资讯、赛事报道等,是一个大型体育综合门户网。赛 V 网主要做体育品牌用品业务,做到只做正品,假一罚十,注重客户服务。这种全新的经营模式为消费者提供了高品质的完美购物体验。

第三种:百货商店

代表:亚马逊、当当网、苏宁网、国美网

商店,说明卖家只有一个;百货,满足日常消费需求的丰富产品线。百货商店有自有仓库,会库存系列产品,以备更快的物流配送和客户服务。这种店甚至会有自己的品牌,如沃

尔玛、屈臣氏等。

3. C2C(Customer to Customer)

C2C 同 B2B、B2C 一样,都是电子商务的几种模式之一。不同的是,C2C 是用户对用户的模式,C2C 商务平台就是通过为买卖双方提供一个在线交易平台,使卖方可以主动提供商品上网拍卖,而买方可以自行选择商品进行竞价。C2C 的典型是百度 C2C、淘宝网、拍拍网等。

电子商务可提供网上交易和管理等全过程的服务,因此它具有广告宣传、咨询洽谈、网上订购、网上支付、电子账户、服务传递、意见征询、交易管理等各项功能。

1) 广告宣传

电子商务可凭借企业的 Web 服务器和客户的浏览,在 Internet 上发布各类商业信息。客户可借助网上的检索工具(Search)迅速地找到所需商品信息,而商家可利用网上主页(HomePage)和电子邮件(E-mail)在全球范围内作广告宣传。与以往的各类广告相比,网上的广告成本最为低廉,而给顾客的信息量却最为丰富。

2) 咨询洽谈

电子商务可借助非实时的电子邮件(E-mail)、新闻组(NewsGroup)和实时的讨论组(Chat)来了解市场和商品信息、洽谈交易事务,如有进一步的需求,还可用网上的白板会议(WhiteboardConference)来交流即时的图形信息。网上的咨询和洽谈能超越人们面对面洽谈的限制,提供多种方便的异地交谈形式。

3) 网上订购

电子商务可借助 Web 中的邮件交互传送实现网上的订购。网上的订购通常都是在产品介绍的页面上提供十分友好的订购提示信息和订购交互格式框。当客户填完订购单后,通常系统会回复确认信息单来保证订购信息的收悉。订购信息也可采用加密的方式使客户和商家的商业信息不会泄露。

4) 网上支付

电子商务要成为一个完整的过程,网上支付是重要的环节。客户和商家之间可采用信用卡账号进行支付。在网上直接采用电子支付手段将可省略交易中很多人员的开销。网上支付将需要更为可靠的信息传输安全性控制以防止欺骗、窃听、冒用等非法行为。

5) 电子账户

网上的支付必须要有电子金融来支持,即银行或信用卡公司及保险公司等金融单位要为金融服务提供网上操作的服务。而电子账户管理是其基本的组成部分。

6) 服务传递

对于已付了款的客户,商家应将其订购的货物尽快地传递到他们的手中。而有些货物在本地,有些货物在异地,电子销售系统将在网络中进行物流的调配。而最适合在网上直接传递的货物是信息产品。

7) 意见征询

电子商务能十分方便地采用网页上的“选择”、“填空”等格式文件来收集用户对销售服务的反馈意见。这样使企业的市场运营能形成一个封闭的回路。客户的反馈意见不仅能提

高售后服务的水平，更使企业获得改进产品、发现市场的商业机会。

8）交易管理

整个交易的管理涉及人、财、物多个方面，涉及企业和企业、企业和客户及企业内部等各方面的协调和管理。因此，交易管理是涉及商务活动全过程的管理。

下面给出开展电子商务的方法和步骤。

第一步：网店命名。名称在网络上是唯一的(具体内容可参考14.4节域名介绍)。

第二步：选择网店地点。网上商店地点的选择要求高速网络连接方式。就好像实体商店位于交通比较方便的地方，用户才可以快速到达(在此步骤中，已经设定好了具体网络地址，客户就可以根据地址访问网店)。

第三步：设计网店的风格。要求进行网页的制作和设计。网上商店同样须装修自己吸引客户。在网页制作中，要求完善后台设计和服务，使得前台和后台程序能够协调运行。

第四步：买卖交易。在网店中，可采用多种方式进行在线交易。比如，第一，买主直接将资金打入卖主银行账户中，凭银行汇款条进行交易；第二，依托支付宝软件，实现第三方支付；第三，网上转账服务等。

第五步：商品运送。网络交易可以通过邮局、大型快递公司，如汇丰、圆通等实现商品的运送。

第六步：商品更新。网页要进行维护，卖出的商品要及时下架，新到货的商品要及时上架。

第七步：库存商品的管理。无论虚拟商店还是真实商店，货物和货币都是一样真实的。对库存商品要及时的清点。

14.7 本章小结

本章针对网站的设计，主要介绍了界面、排版、色彩的设计搭配理论。要求大家熟练掌握这一理论，并且熟练运用元素来搭配自己的网站作品。网站制作的主要流程，先有界面设计，使用 Photoshop 软件；再有网页制作，采用 Dreamweaver 或 Frontpage 软件。实现网站的电子商务功能后，测试网站，运行正确无误后，向域名申请机构申请域名，分配空间，使用 FTP 上传软件将自己的网站上传到服务器中。最后运用推广理论，掌握技巧，推广自己的网站。

14.8 思考与练习

1. 设计出国留学网站界面，并考虑界面、版式、色彩的搭配。
2. 简述域名的申请流程。如何起一个漂亮的域名？
3. 比较 B2B 和 B2C 的优点和缺点。
4. 设计出国留学网站的推广方法。
5. 展开对出国留学网站的电子商务。

第五篇

实训练习

第15章 综合实训练习

15.1 实训1——构建Web服务器

一、实验类别

操作型实验。

二、实验目的

(1) 掌握ASP技术的Web服务器环境。
(2) 熟悉JSP技术的Web服务器环境。

三、实验内容及要求

(1) 熟悉ASP的Web安装环境。
(2) 掌握IIS信息服务器的安装。
(3) 掌握浏览ASP网页的方法。
(4) 掌握创建IIS虚拟目录。
(5) 熟悉JSP的Web安装环境。
(6) 熟悉浏览JSP网页的方法。
(7) 了解SDK的下载及安装。
(8) 了解Tomcat的下载与安装。

四、实验方法及步骤

【题目1】 IIS信息服务器的安装与启动

1. ASP的Web安装环境

在服务器端安装如下软件：Windows 98、Windows 2000 Professional、Windows 2000 Server、Windows 2000 Advance Server、Windows XP Professional或更高版本，其中，Windows 2000系列需要安装Service Pack 2.0；IIS 5.0(Internet信息服务管理器5.0)或更高版本。

客户端只要是普通的浏览器即可，如Internet Explorer 5.0或更高版本。

2. IIS的安装

选择“开始”→“设置”→“控制面板”→“添加/删除程序”命令，在“添加或删除程序”对话

框中单击“添加/删除 Windows 组件”按钮，就会弹出“Windows 组件向导”对话框。在其中选择“Internet 信息服务(IIS)”，然后单击“下一步”按钮，随后根据提示一步步安装即可，如图 15-1 所示。



图 15-1 IIS 的安装

3. IIS 的启动

在“Internet 信息服务”对话框左侧的树窗格中右击“本地计算机”图标，从弹出的快捷菜单中选择“所有任务”→“重新启动 IIS”命令。

浏览 ASP 页面只须右击 ASP 网页，选择“浏览”命令，即可运行，如图 15-2 所示。

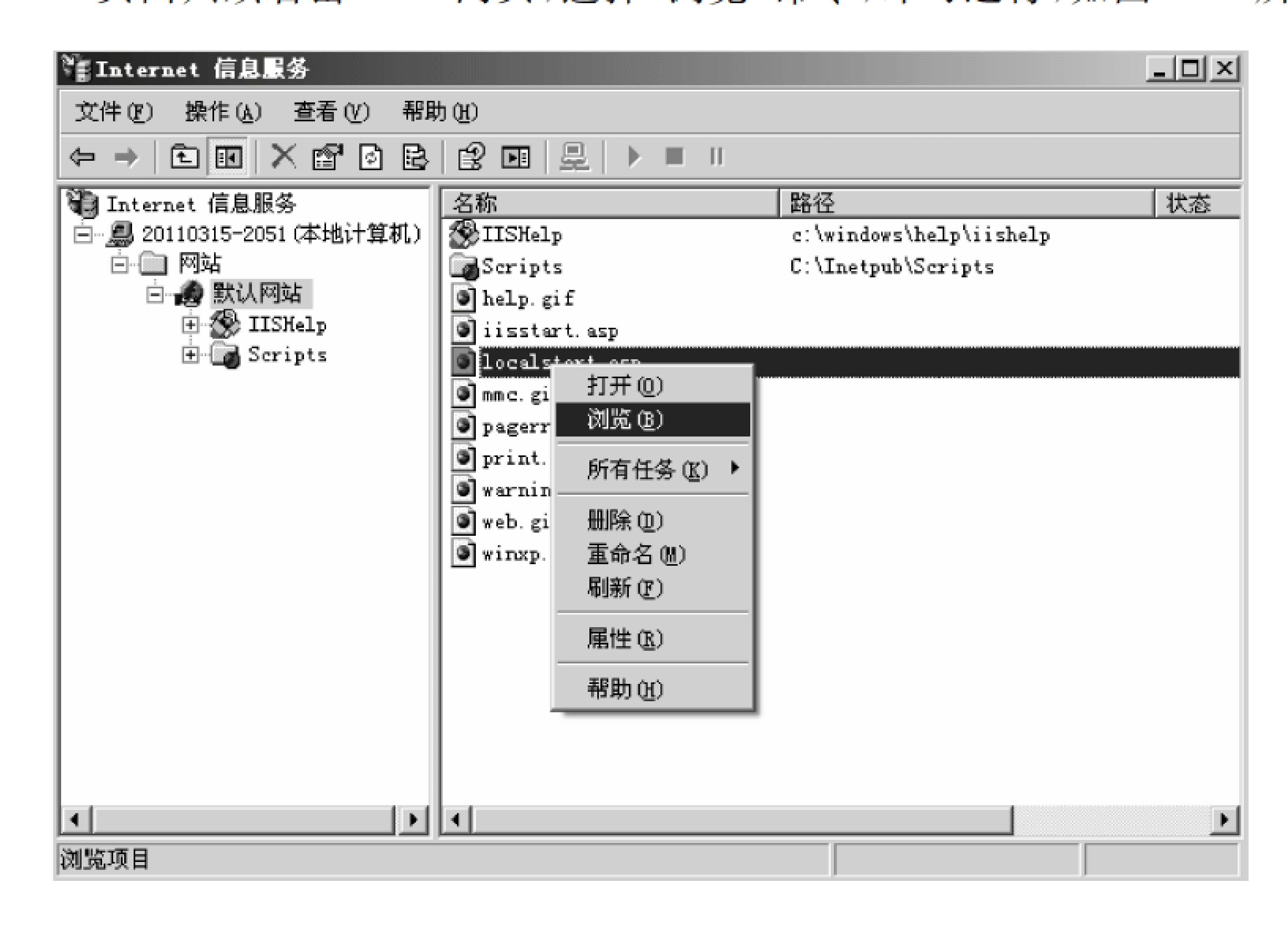


图 15-2 ASP 网页浏览

【题目 2】 创建 IIS 虚拟目录

操作步骤如下：

(1) 启动 IIS 信息服务工具。

(2) 展开本地计算机和网站文件。

(3) 右击“默认网站”，选择“新建”→“虚拟目录”命令，如图 15-3 所示。

(4) 建立虚拟目录别名，例如 online exam，如图 15-4 所示。

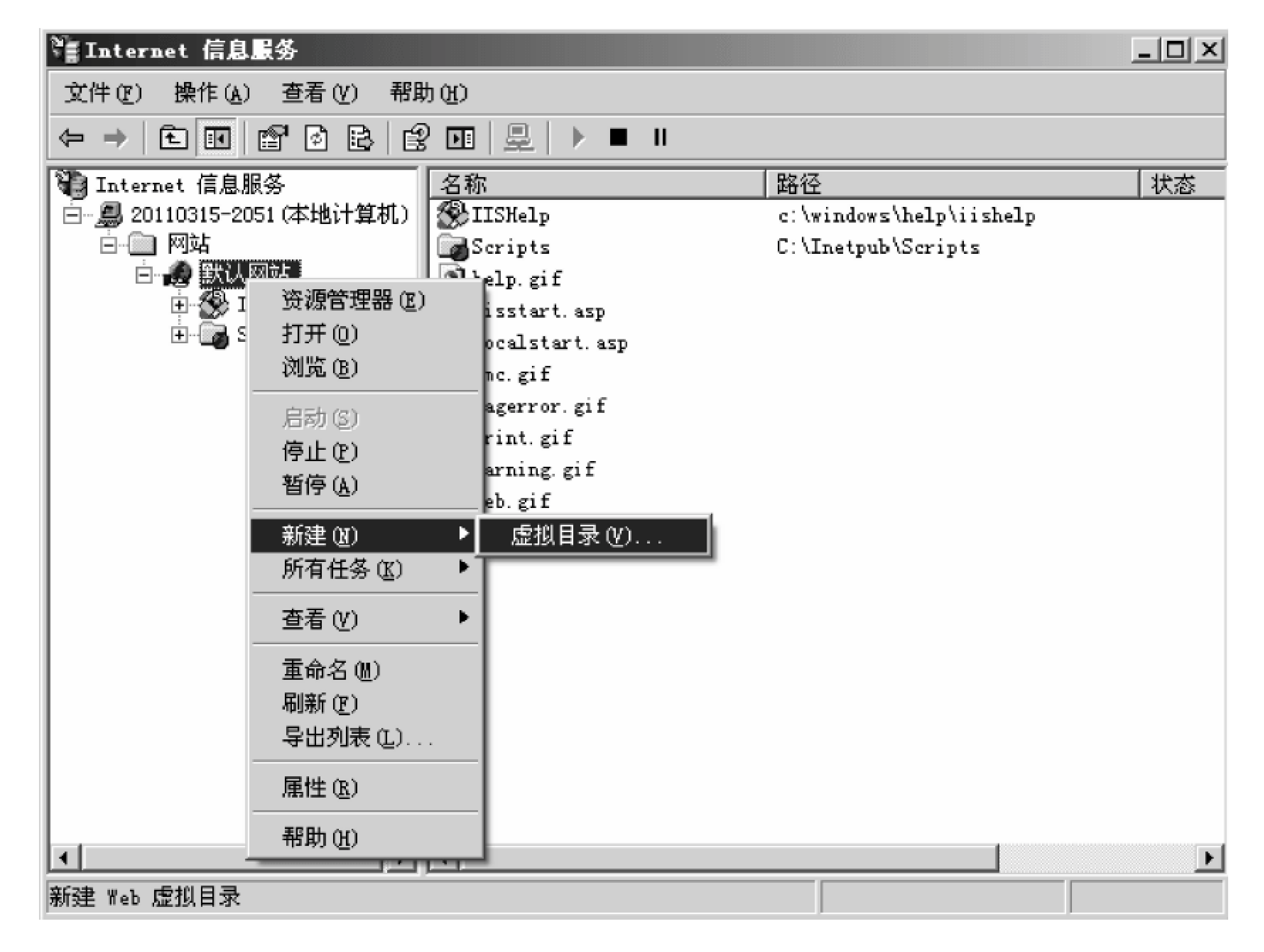


图 15-3 创建虚拟目录

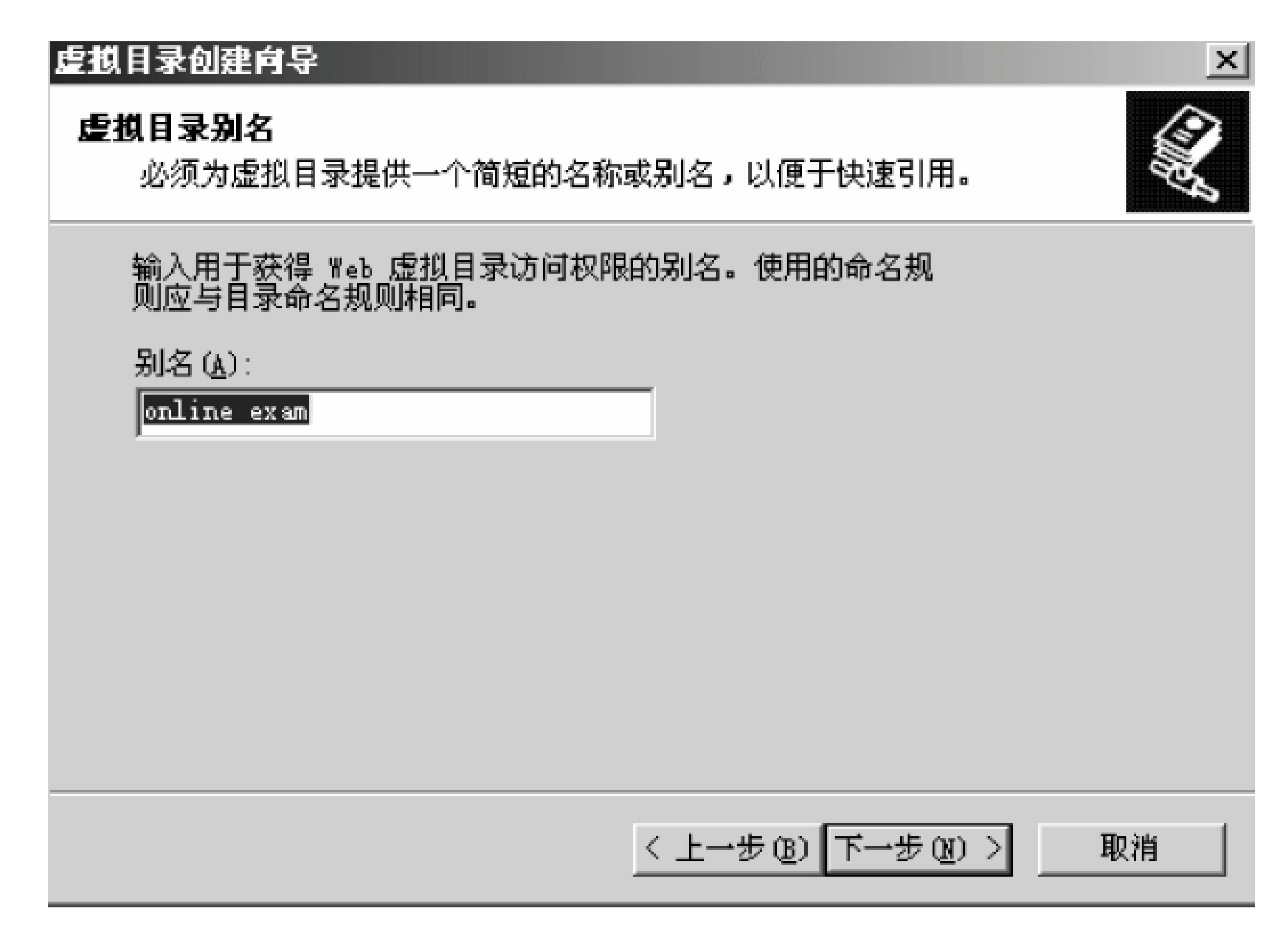


图 15-4 创建虚拟目录别名

（5）输入网站的目录路径，如“E:\在线考试系统”，如图 15-5 所示。



图 15-5 网站的目录路径

【题目 3】 JSP 的 Web 安装环境

JSP 运行环境需要安装两个软件：J2SDK 和 Tomcat。最新的 JDK 为 1.6，Tomcat 为 6.0，建议 JDK 1.4 以上，Tomcat 4.0 以上。

1. JDK 的下载与安装

到 Sun 官方站点(http://java.sun.com/javase/downloads/index.jsp)下载 JDK。

直接双击 JDK 文件，首先出现的是欢迎界面，如图 15-6 所示。选择 JDK 安装路径，如图 15-7 所示。



图 15-6 JDK 欢迎界面



图 15-7　JDK 默认路径

JDK 安装完成界面如图 15-8 所示。

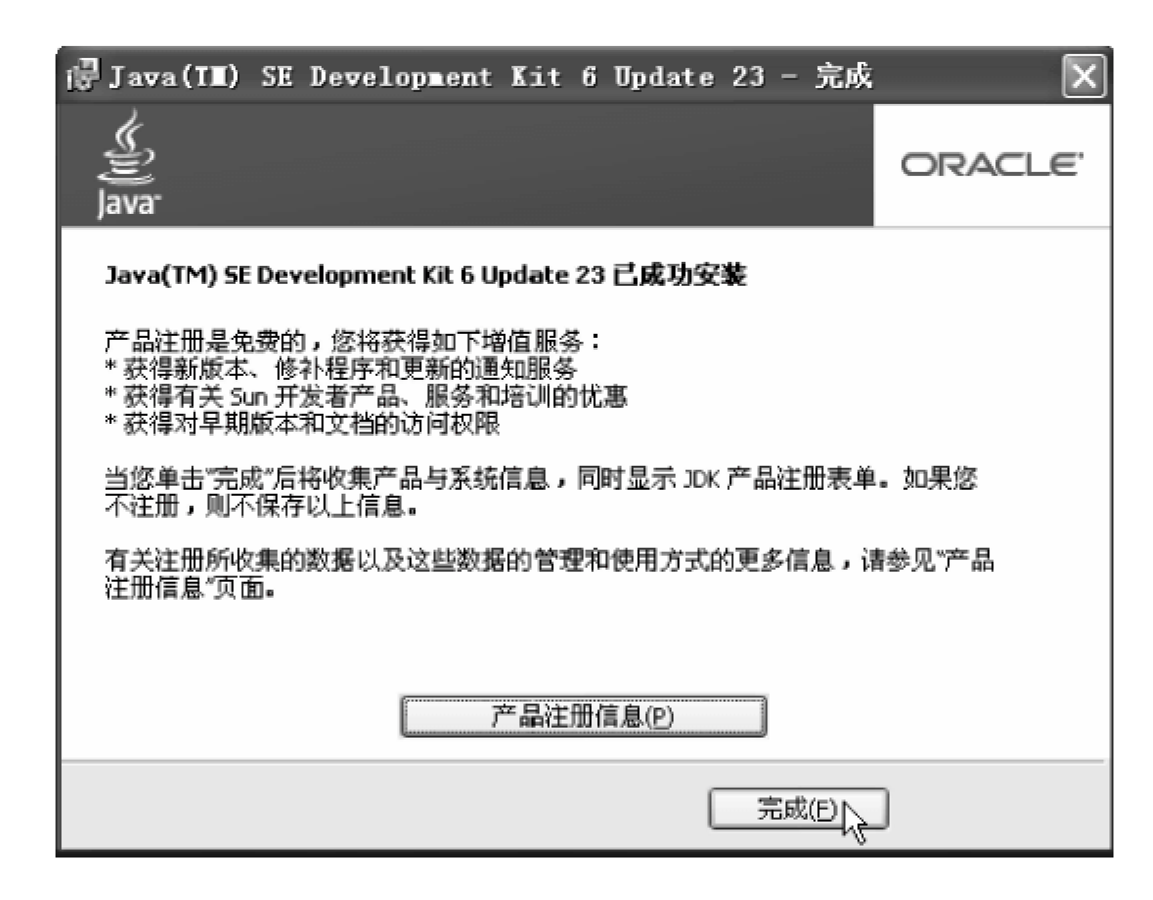


图 15-8　JDK 安装完成界面

安装 J2SDK 以后，需要设置一下环境变量，右击"我的电脑"，选择"属性"→"高级"→"环境变量"，在系统变量中添加环境变量，如图 15-9 所示（假定你的 J2SDK 安装在 D:\Program Files\Java\jdk1.6.0_23\目录下）。

设置内容如下：

```
JAVA_HOME = D:\Program Files\ Java\jdk1.6.0_23\classpath = .; %JAVA_HOME%\lib\dt.jar;%
JAVA_HOME%\lib\tools.jar;(".;"一定不能少,因为它代表当前路径)
path = %JAVA_HOME%\bin
```

接着写一个简单的 Java 程序来测试 J2SDK 是否已安装成功：

```
public class Test{
public static void main(String args[]){
System.out.println("This is a test program.");
    }    }
```



图 15-9 设置 JDK 环境变量

将上面的这段程序保存为文件名为 Test.java 的文件。然后打开命令提示符窗口，执行命令 cd Test.java。

如图 15-10 所示，继续在当前文件所在目录窗口输入下面的命令：

```
javac Test.java
java Test
```

此时如果看到打印出“This is a test program.”，说明安装成功了；如果没有打印出这句话，需要仔细检查一下设置情况，如图 15-10 所示。

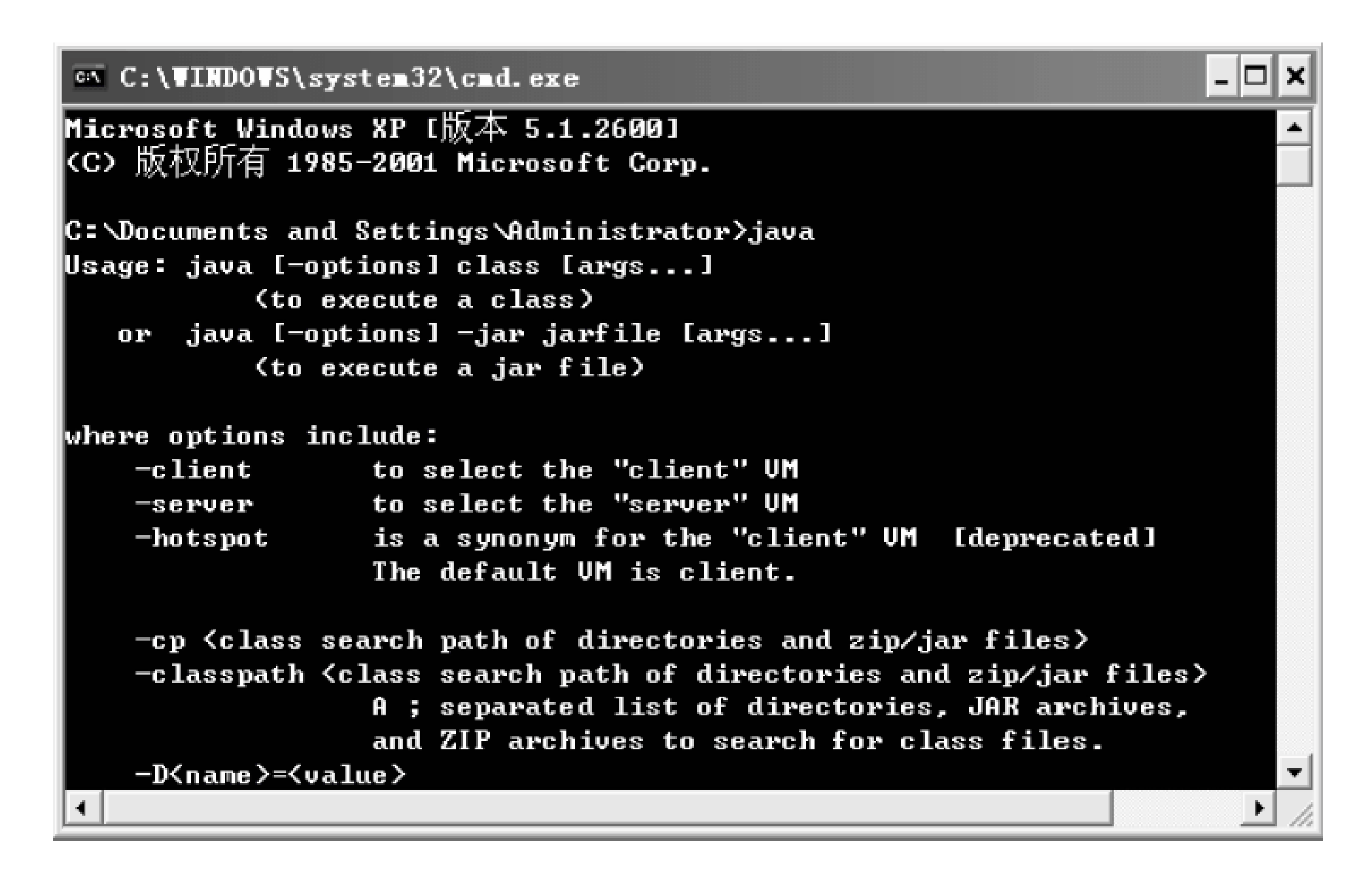


图 15-10 JDK 测试界面

2. Tomcat 的下载与安装

到 Tomcat 官方站点(http://www.apache.org/dist/jakarta/tomcat-4/)下载 Tomcat。安装 Tomcat 后，右击“我的电脑”，选择“属性”→“高级”→“环境变量”，在系统变量中添加

以下环境变量(假定你的 Tomcat 安装在 D:\Program Files\Apache Software Foundation\Tomcat 6.0 目录下)：

```
CATALINA_HOME = D:\Program Files\Apache Software Foundation\Tomcat 6.0
CATALINA_BASE = D:\Program Files\Apache Software Foundation\Tomcat 6.0
```

然后修改环境变量中的 classpath，把 Tomcat 安装目录下的 common\lib 下的(能根据实际追加)servlet.jar 追加到 classpath 中去，修改后的 classpath 如下：

```
classpath = .;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;%CATALINA_HOME%\
common\lib\servlet.jar;
```

接着启动 Tomcat，在 IE 中访问 http://localhost:8080。如果看到 Tomcat 的欢迎页面，说明安装成功了，如图 15-11 所示。

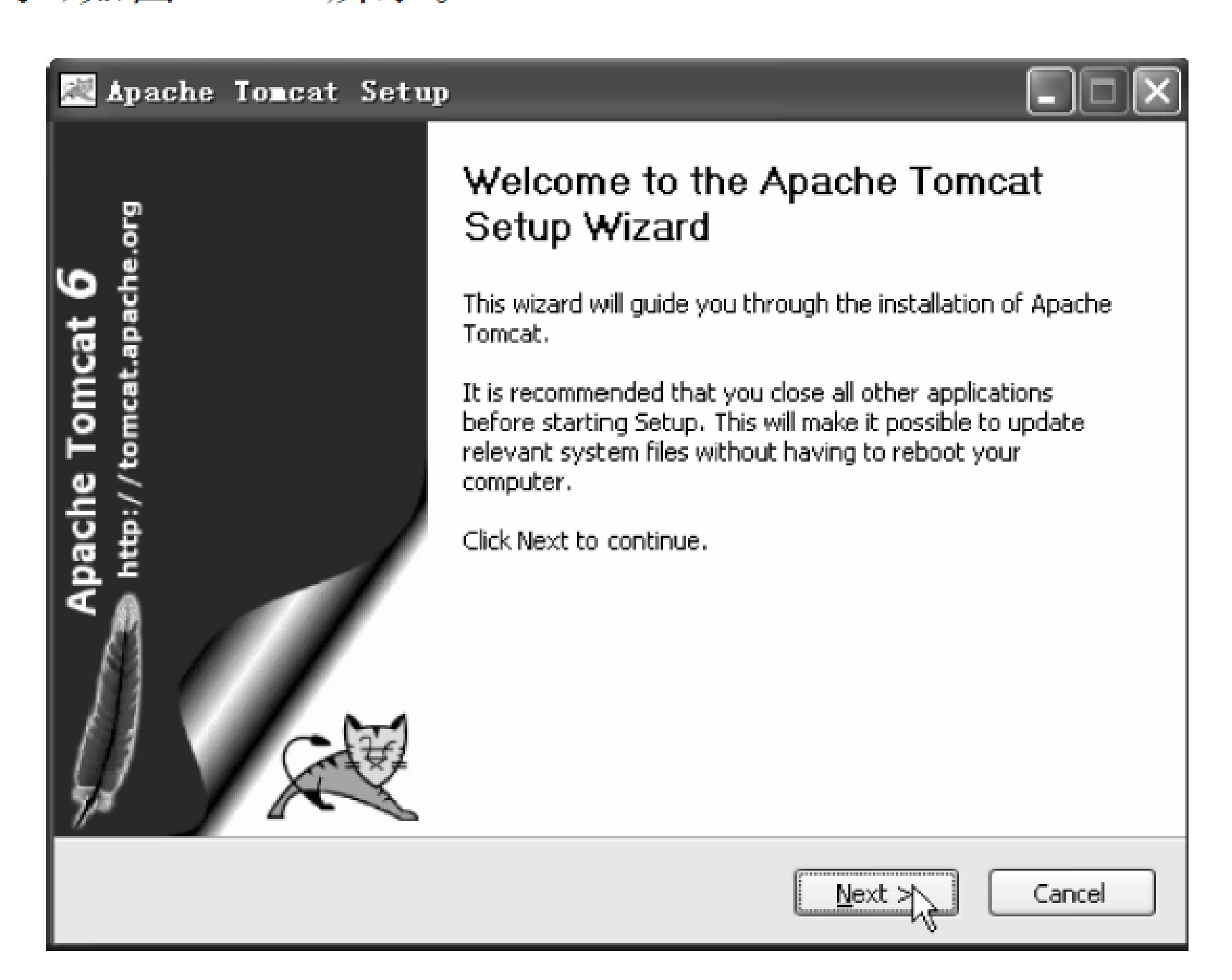


图 15-11　Tomcat 的安装界面

Tomcat 的配置环境界面如图 15-12 所示。

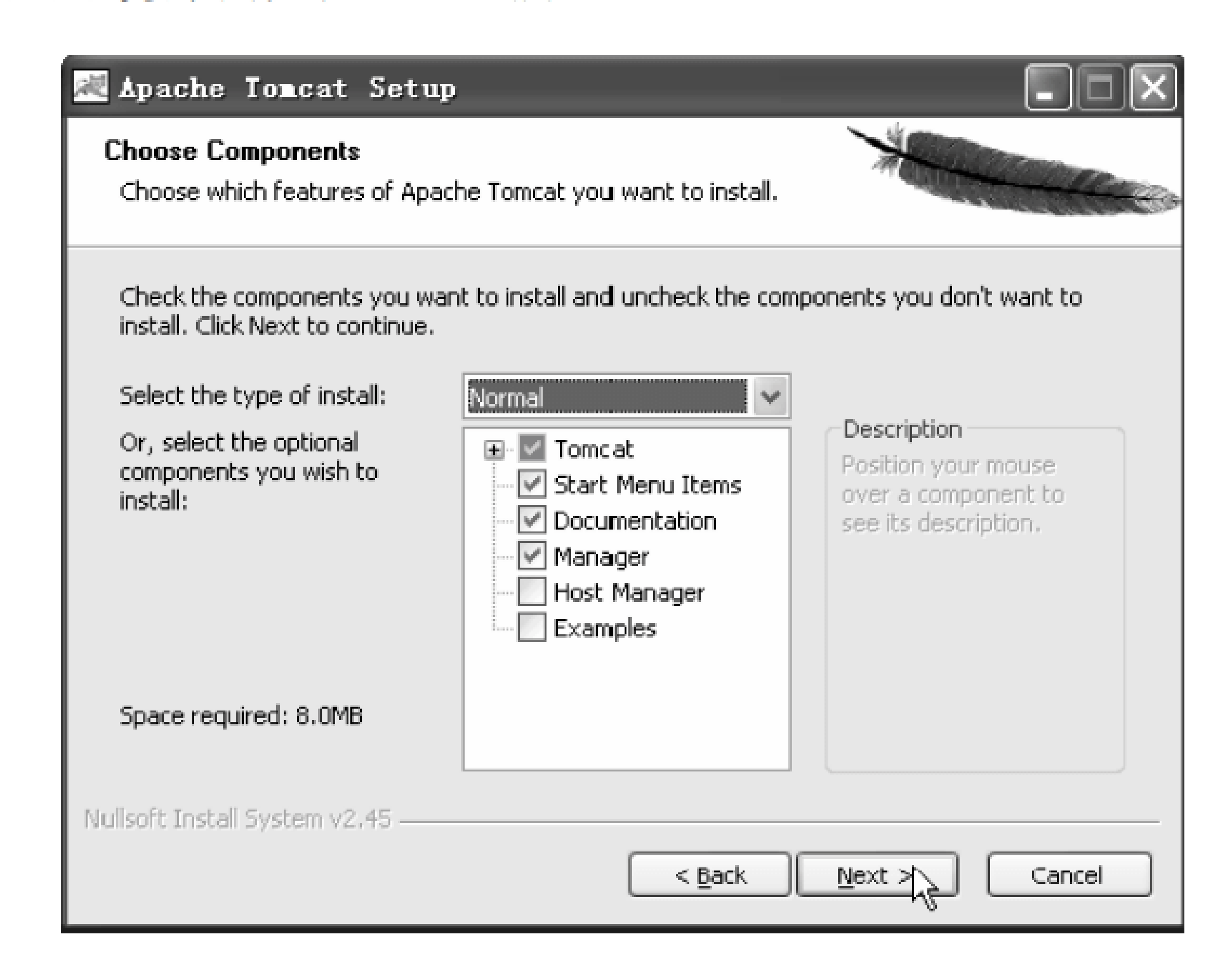


图 15-12　Tomcat 的环境界面

在 Tomcat 参数设置中，需要设置端口号 8080、用户名和登录密码，如图 15-13 所示。

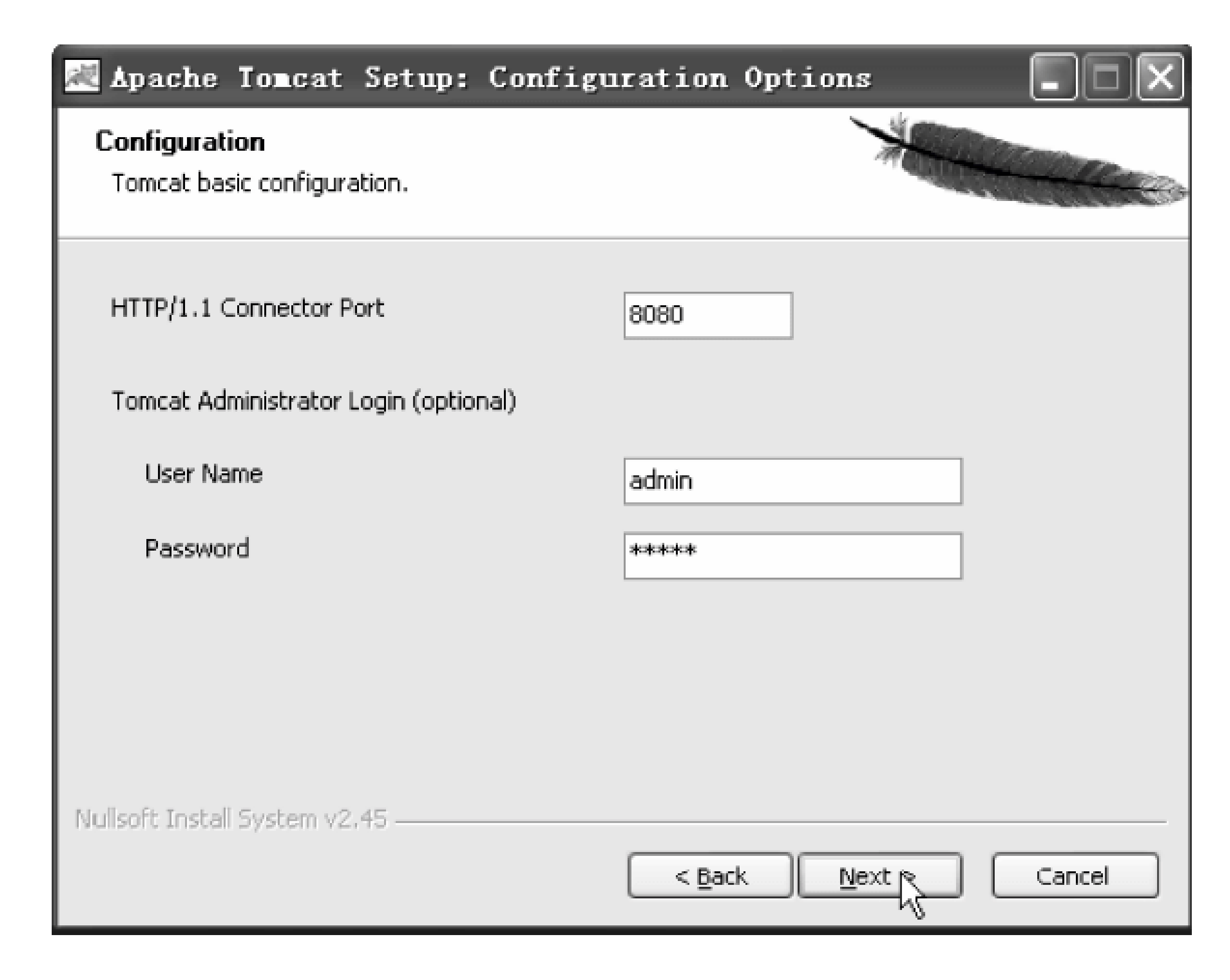


图 15-13 Tomcat 参数设置

Tomcat 安装完成后，显示安装完成界面，如图 15-14 所示。



图 15-14 Tomcat 安装完成界面

在安装完之后，会在计算机显示屏右下角出现 Tomcat 的运行图标，如图 15-15 所示。



图 15-15 Tomcat 服务器运行

3. 运行 JSP 网页

(1) 到 Tomcat 的安装目录下的 webapps 目录中能看到 ROOT、examples、tomcat-docs 之类 Tomcat 自带的目录。

(2) 在 webapps 目录下新建一个目录，命名为 myapp。

(3) 在 myapp 下新建一个目录 WEB-INF，注意，目录名称是区分大小写的。

(4) 在 WEB-INF 下新建一个文件 web.xml,内容如下:

```
My Web Application
A application for test.
```

(5) 在 myapp 下新建一个测试的 JSP 页面,文件名为 index.jsp,文件内容如下:

```
Now time is:
```

(6) 重启 Tomcat。

(7) 打开浏览器,输入"http://localhost:8080/myapp/index.jsp",如果看到当前时间,则说明运行成功了。

五、操作技巧与注意事项

(1) 注意 IIS 虚拟路径的设置方法。
(2) 注意 JDK 路径的设置地址,并在设置环境变量时能够正确测试出结果。
(3) 能够使用 Tomcat 服务测试 JSP 页面。
(4) 能够使用 IIS 服务测试 ASP 页面。

六、思考题与练习

(1) 练习使用在线考试网站(ASP),并测试整个网站。
(2) 练习并测试物资管理网站(JSP),能够正确运行,并查看结果。

15.2 实训 2——实现 HTML 网页

一、实验类别

操作型实验。

二、实验目的

(1) 熟练掌握 HTML 标记的使用方法。
(2) 能够制作五洲留学网站的界面。

三、实验内容及要求

(1) 熟练掌握 HTML 页面标记。
(2) 熟练运用文档格式标记、字号、文本修饰、段落格式。
(3) 实现超链接标记。
(4) 能够设计表单页面。
(5) 熟练使用 CSS 样式设计页面。

四、实验方法及步骤

本实训通过建立五洲留学网站，掌握对 HTML 网页的制作。以加拿大网站页面为例，讨论制作方法。

软件：Dreamweaver MX 以上版本、Windows XP。

操作方法：创建 3 个页面文件，分别是 top. htm、bottom. htm、index. htm。

1. top.htm 文件的制作方法

打开 Dreamweaver 软件，在页面上右击，选择“页面属性”，将左边距和上边距设置为 0，然后按照图 15-16 创建表格(4 行 1 列)。

top. htm 文件的界面如图 15-16 所示。

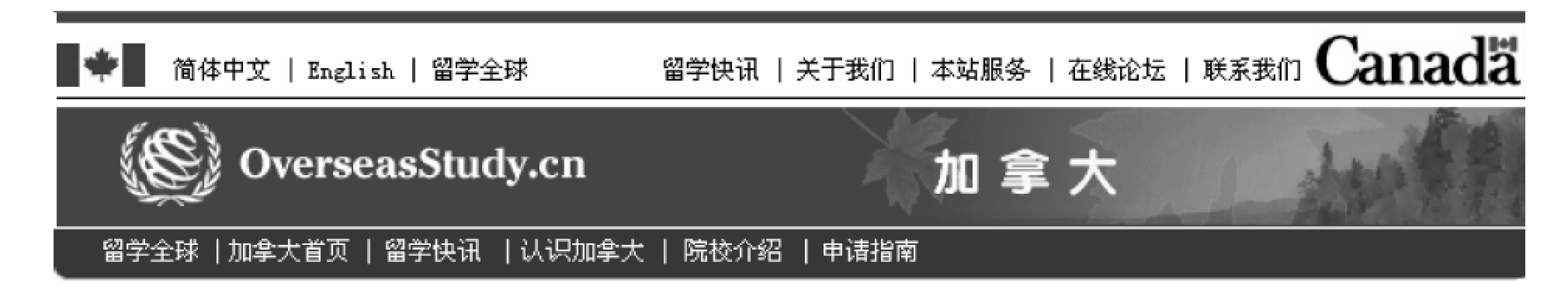


图 15-16　top. htm 界面

【top. htm 代码】

```
<HTML>
<HEAD>
<META http-equiv = "Content-Type" content = "text/html; charset = gb2312">
<TITLE>留学快讯</TITLE>
<META name = "GENERATOR" content = "Microsoft FrontPage 3.0">
<LINK rel = "stylesheet" href = "public/style.css">
</HEAD>
<BODY topmargin = "0" leftmargin = "0">
<DIV align = "center"><CENTER>
<TABLE border = "0" cellpadding = "0" cellspacing = "0" width = "715">
  <TR>
    <TD height = "5" bgcolor = "#B0151B"></TD>
  </TR>
  <TR>
    <TD valign = "middle" height = "29" bgcolor = "#FFFFFF"><DIV align = "center"><CENTER><
TABLE border = "0" cellpadding = "0" cellspacing = "0" width = "715">
        <TR>
          <TD width = "45"><IMG src = "images/logo_002.jpg" width = "42" height = "29"></TD>
          <TD width = "626"><DIV align = "center"><CENTER><TABLE border = "0" cellpadding = "0"
          cellspacing = "0" width = "100%" height = "19">
            <TR>
              <TD height = "15"></TD>
            </TR>
            <TR>
              <TD width = "100%" height = "14"><DIV align = "center"><CENTER><TABLE border =
```

```
"0" cellpadding = "0"
            cellspacing = "0" width = "100 %">
              < TR >
                < TD width = "36 % " valign = "bottom">   < A HREF = "index. htm" class =
"b1">简体中文</A > | < A
                HREF = "../en/index. htm" class = "b1"> English </A > | < A HREF = "../../index.
htm" class = "b1">留学全球</A ></TD >
                < TD width = "8 % " valign = "bottom"></TD >
                < TD width = "56 % " valign = "bottom">< A HREF = "../../news/index. htm" class =
"b1">留学快讯</A >
                | < A HREF = "../../about_us. htm" class = " b1 ">关于我们</A > | < A HREF =
"../../serve. htm"
                class = "b1">本站服务</a > | < a href = "../../bbs. htm" class = "b1">在线论坛
</A > | < A HREF = "../../contact_us. htm" class = "b1">联系我们</A ></TD >
              </TR >
            </TABLE >
            </CENTER ></DIV ></TD >
          </TR >
        </TABLE >
        </CENTER ></DIV ></TD >
        < TD width = "103">< img src = "images/logo_003. jpg" width = "101" height = "29"></TD >
      </TR >
    </TABLE >
    </CENTER ></DIV ></TD >
  </TR >
  < TR >
    < TD valign = "top" height = "3" bgcolor = "#FFFFFF"></TD >
  </TR >
  < TR >
    < TD valign = "top" height = "1" bgcolor = "#000000"></TD >
  </TR >
  < TR >
    < TD valign = "top" height = "4" bgcolor = "#FFFFFF"></TD >
  </TR >
</TABLE >
</CENTER ></DIV >< DIV align = "center">< CENTER >

< TABLE border = "0" cellpadding = "0" cellspacing = "0" width = "715">
  < TR >
    < TD height = "57" valign = "top">< DIV align = "center">< CENTER >< TABLE border = "0"
cellpadding = "0"
    cellspacing = "0" height = "57" width = "715" bgcolor = "#B0151B">
      < TR >
        < TD width = "37"></TD >
        < TD width = "407">< IMG src = "images/logo_001. jpg" width = "320" height = "51"></TD >
        < TD width = "334">< IMG src = "images/pic_001. jpg" width = "332" height = "58"></TD >
      </TR >
    </TABLE >
    </CENTER ></DIV ></TD >
  </TR >
  < TR >
```

```
    <TD height="1" valign="top" width="715"></TD>
  </TR>
  <TR>
    <TD height="20" valign="middle" bgcolor="#363636" width="679"><DIV align=
"center"><CENTER><TABLE
    border="0" cellpadding="0" cellspacing="0" width="717">
      <TR>
        <TD width="24"><IMG src="images/pic_002.jpg" width="5" height="24"></TD>
        <TD width="59" class="f2"><A HREF="../../index.htm">留学全球</A></TD>
        <TD width="10" class="f2"><FONT color="#FFFFFF">|</FONT></TD>
        <TD width="73" class="f2"><A HREF="index.htm">加拿大首页</A></TD>
        <TD width="12" class="f2"><FONT color="#FFFFFF">|</FONT></TD>
        <TD width="64" class="f2"><A HREF="news.htm">留学快讯</A></TD>
        <TD width="11" class="f2"><FONT color="#FFFFFF">|</FONT></TD>
        <TD width="74" class="f2"><A HREF="index.htm">认识加拿大</A></TD>
        <TD width="14" class="f2"><FONT color="#FFFFFF">|</FONT></TD>
        <TD width="62" class="f2"><A HREF="college.htm">院校介绍</A></TD>
        <TD width="12" class="f2"><FONT color="#FFFFFF">|</FONT></TD>
        <TD width="60" class="f2"><A HREF="application.htm">申请指南</A></TD>
        <TD width="316"></TD>
        <TD width="16"><DIV align="center"><CENTER><TABLE border="0" cellpadding="0"
        cellspacing="0" width="113%">
          <TR>
            <TD width="63%"></TD>
            <TD width="50%"><IMG src="images/pic_003.jpg" width="6" height="24"></TD>
          </TR>
        </TABLE>
        </CENTER></DIV></TD>
      </TR>
    </TABLE>
    </CENTER></DIV></TD>
  </TR>
  <TR>
    <TD height="1" valign="top" width="715" bgcolor="#C8C8C8"><DIV align="center"><
CENTER><TABLE
    border="0" cellpadding="0" cellspacing="0" width="715">
      <TR>
        <TD width="9"><DIV align="center"><CENTER><TABLE border="0" cellpadding="0"
        cellspacing="0" width="100%">
          <TR>
            <TD width="46%" bgcolor="#ffffff"></TD>
            <TD width="54%"></TD>
          </TR>
        </TABLE>
        </CENTER></DIV></TD>
        <TD width="66"></TD>
        <TD width="12"></TD>
        <TD width="57"></TD>
        <TD width="11"></TD>
        <TD width="58"></TD>
        <TD width="13"></TD>
```

```
<TD width="60"></TD>
<TD width="11"></TD>
<TD width="61"></TD>
<TD width="11"></TD>
<TD width="67"></TD>
<TD width="337"></TD>
<TD width="16"><DIV align="center"><CENTER><TABLE border="0" cellpadding="0"
cellspacing="0" width="113%">
  <TR>
    <TD width="63%"></TD>
    <TD width="50%" bgcolor="#ffffff"></TD>
  </TR>
</TABLE>
</CENTER></DIV></TD>
</TR>
</TABLE>
</CENTER></DIV></TD>
</TR>
<TR>
<TD height="1" valign="top" width="715" bgcolor="#EDEDED"><DIV align="center">
<CENTER><TABLE
  border="0" cellpadding="0" cellspacing="0" width="715">
  <TR>
    <TD width="9"><DIV align="center"><CENTER><TABLE border="0" cellpadding="0"
    cellspacing="0" width="100%">
      <TR>
        <TD width="46%" bgcolor="#ffffff"></TD>
        <TD width="54%"></TD>
      </TR>
    </TABLE>
    </CENTER></DIV></TD>
    <TD width="66"></TD>
    <TD width="12"></TD>
    <TD width="57"></TD>
    <TD width="11"></TD>
    <TD width="58"></TD>
    <TD width="13"></TD>
    <TD width="60"></TD>
    <TD width="11"></TD>
    <TD width="61"></TD>
    <TD width="11"></TD>
    <TD width="67"></TD>
    <TD width="337"></TD>
    <TD width="16"><DIV align="center"><CENTER><TABLE border="0" cellpadding="0"
    cellspacing="0" width="113%">
      <TR>
        <TD width="63%"></TD>
        <TD width="50%" bgcolor="#ffffff"></TD>
      </TR>
    </TABLE>
    </CENTER></DIV></TD>
```

```
        </TR>
      </TABLE>
      </CENTER></DIV></TD>
    </TR>
  </TABLE>
  </CENTER></DIV>
  </BODY>
  </HTML>
```

2. bottom.htm 文件的制作方法

打开 Dreamweaver 软件，右击鼠标，选择“页面属性”，将左边距和上边距设置为 0。然后按照图 15-17 创建表格(2 行 1 列)。

bottom.htm 文件的界面如图 15-17 所示。

ENGLISH 政策法规 常用链接 寻求合作 广告服务 友情链接 版权声明 隐私权保护

Copyright 2004 Overseasstudy.cn All Rights Reserved

图 15-17　bottom.htm 文件

【bottom.htm 代码】

```
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=gb2312">
<TITLE>ENGLISH 政策法规 常用链接 寻求合作 广告服务 友情链接
  版权声明 隐私权保护</TITLE>
<META name="GENERATOR" content="Microsoft FrontPage 3.0">
</HEAD>
<BODY topmargin="0" leftmargin="0">
<DIV align="center"><CENTER>
<TABLE border="0" cellpadding="0" cellspacing="0" width="715">
  <TR>
    <TD valign="middle" height="22" bgcolor="#363636" align="center"><P align=
"center"><FONT
    size="2" color="#FFFFFF"><A HREF="../en/index.htm">ENGLISH</A><A HREF="../../
policy.htm">政策法规</A>
    <A HREF="../../common_links.htm">常用链接</A><A HREF="../../cooperate.htm">寻求合
作</A>
    <A HREF="../../ad/index.htm">广告服务</A><A HREF="../../links.htm">友情链接
</A><A
    HREF="../../copyright.htm">版权声明</A><A HREF="../../contact_us.htm">隐私权保护
</A></FONT></TD>
  </TR>
  <TR>
<TD valign="top" height="2"></TD>
</TR>
  <TR>
    <TD height="61" valign="middle" bgcolor="#B0151B"><DIV align="center"><CENTER>
```

```
<TABLE
    border="0" cellpadding="0" cellspacing="0" width="361">
      <TR>
      <TD width="361"><FONT face="Verdana" size="2" color="#FFFFFF">Copyright 2004
        Overseasstudy.cn All Rights Reserved</FONT></TD>
      </TR></TABLE></CENTER></DIV></TD>
  </TR></TABLE></CENTER></DIV></BODY></HTML>
```

3. index.htm 文件的制作方法

index.htm 主页文件中包含 3 个文件，分别是 top.htm、bottom.htm、index.htm。其中，top.htm 和 bottom.htm 是独立的外部文件，须通过 JavaScript 代码将两个文件嵌入到 index.htm 主页文件中。合成后的 index.htm 主页文件如图 15-18 所示。

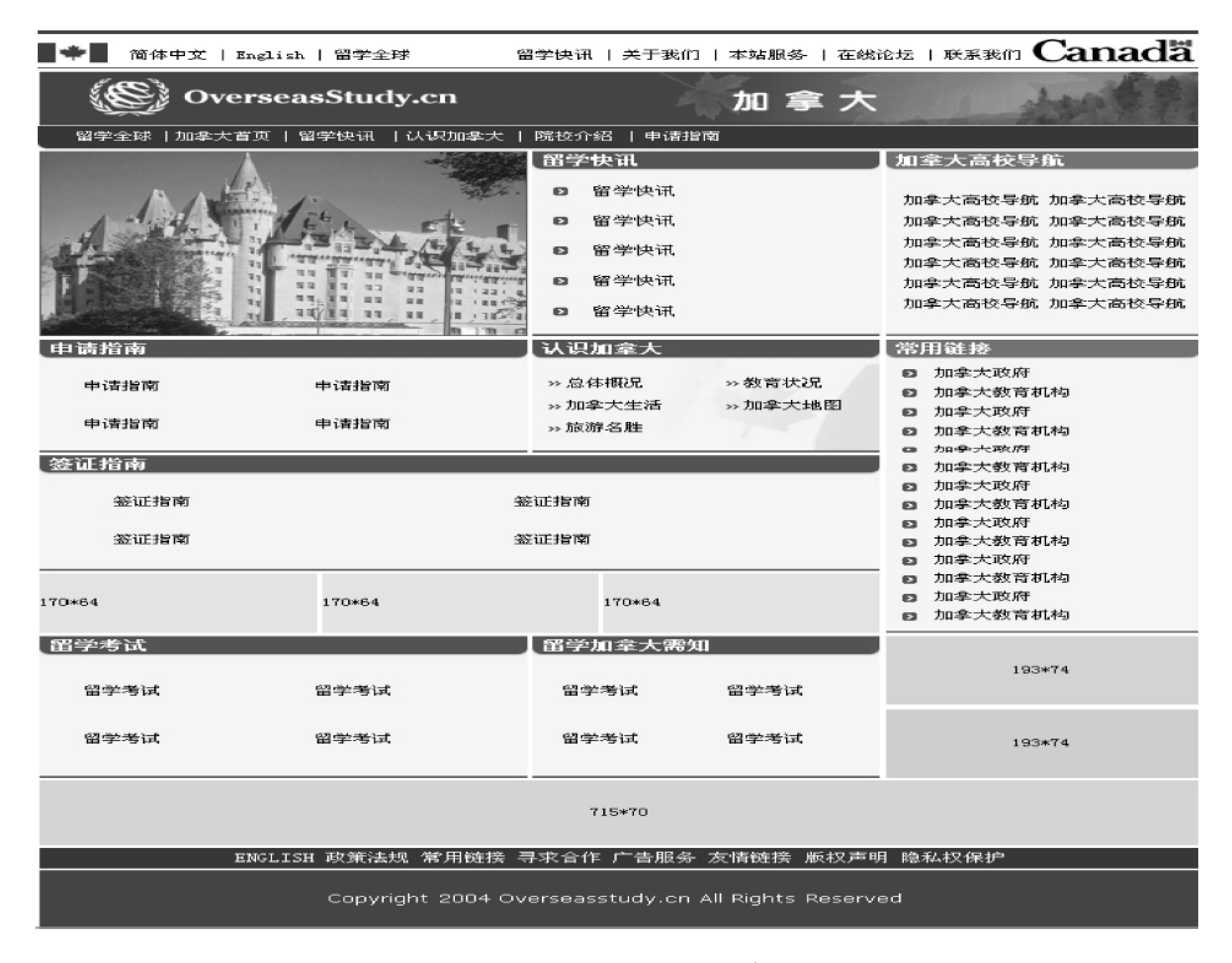


图 15-18 index.htm 文件

在 index.htm 中加入如下 JavaScript 代码：

```
<SCRIPT>
    function onDownloadDone(downDate){              //定义页面事件触发时加载的函数
    showImport.innerHTML = downDate                 //加载嵌入 htm 网页的方法
    }
    oDownload.startDownload('top.htm',onDownloadDone)     //嵌入文件名 top.htm
      </SCRIPT>
```

```
<SCRIPT>
    function onDownloadDone1(downDate){              //定义页面事件触发时加载的函数
    showImport1.innerHTML = downDate                 //加载嵌入 htm 网页的方法
    }
oDownload1.startDownload('bottom.htm',onDownloadDone1)  //嵌入文件名 bottom.htm
</SCRIPT>
```

4. CSS 样式的控制

网页中使用 CSS 样式控制网页的字体、图片、链接的样式。

1）字体样式控制

```
.f1 {
  COLOR: #ffffff;                //颜色样式
  FONT: 14px 宋体;               //字体大小和字体形状
font-weight:bold;                //字体加粗
line-height:18px   }             //行距设置 18 像素
```

2）链接样式控制

```
a.a1:link {                      //鼠标访问前样式
  COLOR: #ffffff;
  TEXT-DECORATION: none}
a.a1:visited {                   //鼠标访问后样式
  COLOR: #ffffff;
  TEXT-DECORATION: none }
a.a1:active {                    //鼠标激活后样式
  COLOR: #000000;
  TEXT-DECORATION: none}
a.a1:hover {                     //鼠标浮动时样式
  COLOR: #ffffff;
  TEXT-DECORATION: underline}
```

在使用 CSS 控制中，操作方法分为两个步骤。

第一步：在 head 标记中加入<link rel="stylesheet" href="public/style.css">代码，如下所示：

```
<HEAD>
<META http-equiv = "Content-Type" content = "text/html; charset = gb2312">
<TITLE>五洲留学网 &middot; 留学全球 www.overseasstudy.cn</TITLE>
<META name = "GENERATOR" content = "Microsoft FrontPage 3.0">
<LINK rel = "stylesheet" href = "public/style.css">
</HEAD>
```

第二步：在对应控制的文本处标注 class="f1"，代码如下所示：

```
<TD width = "202" bgcolor = "#C12424" height = "19" class = "f1" valign = "bottom">
<A HREF = "news.htm">留学快讯</A>
```

五、操作技巧与注意事项

(1) 使用 CSS 样式控制网页文本、链接字体。
(2) 使用 JavaScript 技术将 top. htm、bottom. htm 和 index. htm 合成在一起。
(3) 注意表格和图片的代码设计。
(4) 注意文本和图片链接方法。

六、思考题与练习

(1) 制作 top. htm 文件的方法。
(2) 制作 bottom. htm 文件的方法。
(3) 制作 index. htm 文件的方法。
(4) 自行设计一个字体控制样式 f2,然后根据此样式,应用到 index. htm 页面中。

15.3 实训 3——JavaScript 和 VBScript 脚本语言的实验

一、实验类别

操作型实验。

二、实验目的

(1) 掌握 VBScript 脚本的使用。
(2) 熟练掌握 JavaScript 脚本的使用。

三、实验内容及要求

实验软件:Dreamweaver、IIS。
VBScript 实验:
(1) 显示当月日历:把当月所有天数的数字输出,并按标准日历形式排列。
(2) 下拉列表的实验:动态产生下拉列表。
JavaScript 实验:注册页面的实验。

四、实验方法及步骤

1. 显示当月日历

把当月所有天数的数字输出,并按标准日历形式排列。

显示当月日历代码运行结果如图 15-19 所示,这是一个月的日历,并显示星期。在这段代码中,weekday()函数返回代表一星期中某天的整数(1~7),分别表示星期日至星期六;CDate()函数将字符串转换成日期类型。



图 15-19　显示当前日历

首先用 if i<week then 语句判断每月的第一天是星期几，如果 i 表示的数字小于星期表示的数字，一直向后移，直到 i=week，例如本例中 3 月份的第一天是星期六，第一行的前 6 个表格用空来表示，直接把 1 输入到星期六对应的表格中。

循环语句 do…loop 分别把其他的数字与表示年份和月份的数字组合，然后用 IsDate() 函数判断是不是本月合法的日期，如果是则输出。

操作代码：

```
…//省略 HTML 代码
<TR>
<TD class = "gray">日</TD><TD class = "gray">一</TD><TD class = "gray">二</TD>
<TD class = "gray">三</TD><TD class = "gray">四</TD><TD class = "gray">五</TD>
<TD class = "gray">六</TD></TR>
<%
date_time = date() '获取当前系统日期并赋值给变量 date_time
r_year = year(date_time) '取得当前年份并赋值给 r_year
r_month = month(date_time) '取得当前月份并赋值给 r_month
week = weekday(cdate(r_year&" - "&r_month&" - 1"))
'当月第一天表示星期的整数并赋值给变量 week
response.write "<TR align = center>"  '输出一行
i = 1 'i 初始值设为 1
today = 1 'today 初始值设为 1
do while today <= 31 '当 today 值等于 31 时,结束循环
   If i < week then '如果 i 值小于星期的整数时,输出一列单元格
     Response.Write "<TD> </TD>"
     i = i + 1
   Else
     If isdate(r_year&" - "&r_month&" - "&today) then '判断是不是有效日期
         Response.Write "<TD>"&today&"</TD>"
     End If
     If i mod 7 = 0 then '判断是不是一周的最后一天,如果是则换行
        Response.Write "</TR><TR align = center>"
     End if
        i = i + 1
     today = today + 1
```

```
    End If
loop
%>
</TABLE></BODY></HTML>
```

2. 下拉列表

操作方法：

首先创建一个 ASP 文件，从表单中加入下拉列表控件，如图 15-20 所示，然后定义数组 array，并设定各省的名称，最后用 response. write 方法进行输出。

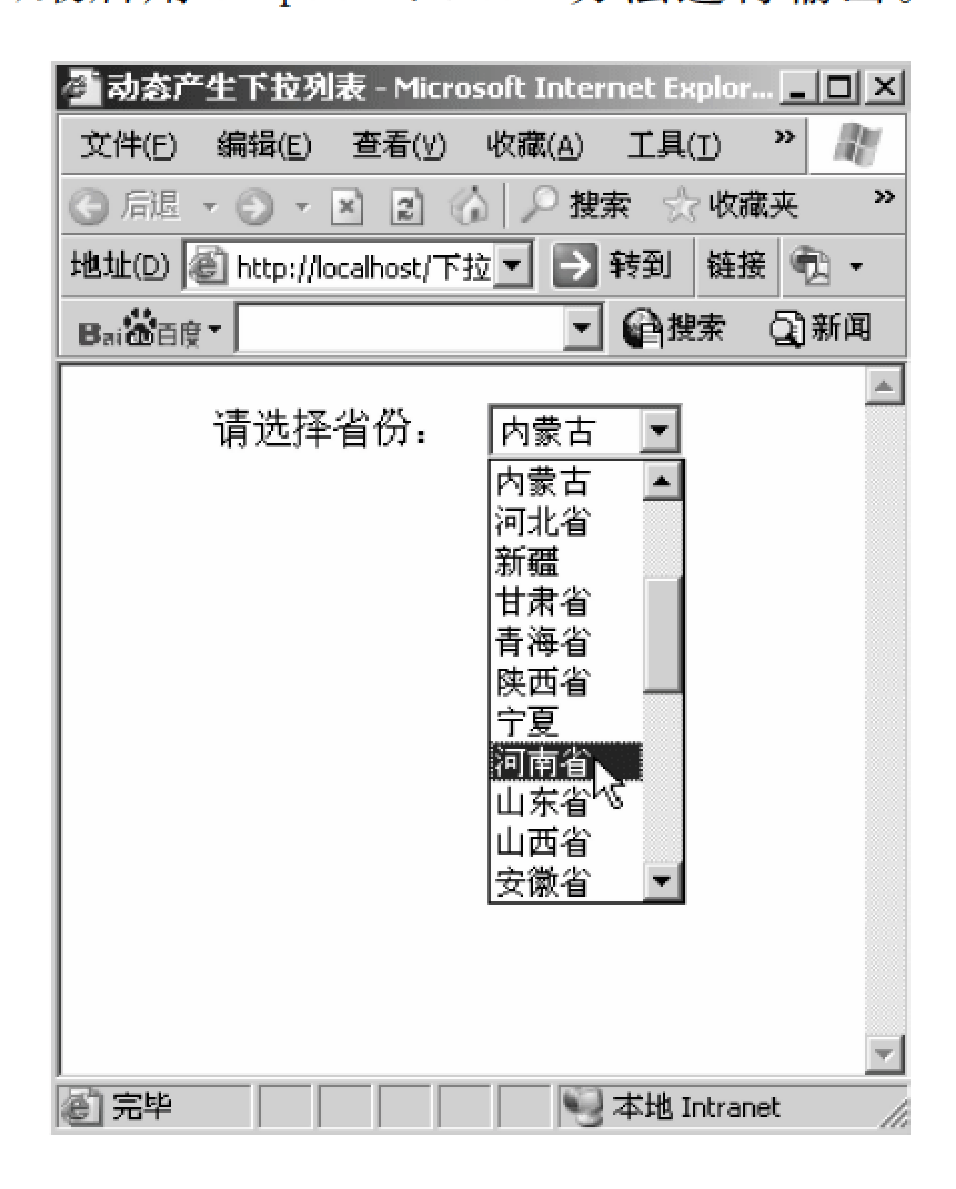


图 15-20　下拉列表

操作代码：

```
<HTML xmlns = "http://www.w3.org/1999/xhtml">
<HEAD>
<META http - equiv = "Content - Type" content = "text/html; charset = utf - 8" />
<TITLE>动态产生下拉列表</TITLE>
</HEAD>
<BODY>
<TABLE width = "200" border = "0" cellspacing = "0" cellpadding = "0" align = "center">
  <TR>
    <TD>请选择省份：</TD>
    <TD><SELECT
name = "province">
<% dim province, i
province = array("北京市","上海市","天津市","重庆市","黑龙江省","吉林省","辽宁省","内蒙古","河北省","新疆","甘肃省","青海省","陕西省","宁夏","河南省","山东省","山西省","安徽省","湖北省","湖南省","江苏省","四川省","贵州省","云南省","广西","西藏","浙江省","江西省","广东省","福建省","台湾省","海南省","香港","澳门")
for each  i in province
```

```
Response.write("<option
value="&i&">"&i&"</option>") next
%>
</SELECT></TD>
        </TR></TABLE></BODY></HTML>
```

3. 注册页面的实验

操作方法：创建一个 ASP 文件，从表单中加入文本框控件、单选按钮、下拉列表、图像控件、复选框控件、按钮控件，按照图 15-21 位置排列，然后加入 JavaScript 脚本代码进行验证。

JavaScript 脚本验证代码：

```
<SCRIPT Language="javascript">
function checkNull(txtItem,txtName){          //参数分别表示文本框对象及其对应中文
  if (txtItem.value == "")                    //若文本框对象的值为空
  {
  alert(txtName + "不能为空");                //进行提醒
  txtItem.focus();                            //将光标焦点指向该文本框
  return false;                               //函数返回值为假
  }else return true;                          //否则表示填写过内容,函数返回值为真
  }
function checkItem(itemObj,itemName){         //参数分别表示单选框对象及其对应中文
  var flag = false;
for(var i = 0;i<itemObj.length;i++){          //循环检查单选框对象是否被选中
if(itemObj[i].checked){
  flag = true;}                               //做标记
  }
  if(flag)                                    //若被选中
    return true;
  else{                                       //未被选中
    alert(itemName + "不能为空");             //提示
    return false;
  }
}
function checkForm(){                         //验证表单元素是否为空
    if (
      checkNull(document.getElementById("uid"),"用户名")&&
      checkNull(document.getElementById("pwd"),"密码")&&
      checkItem(document.all.sex,"性别")&&
      checkNull(document.getElementById("userFace"),"头像")&&
      checkNull(document.getElementById("email"),"电子邮件")&&
      checkItem(document.all.character,"性格")&&
      checkItem(document.all.interest,"兴趣爱好")
     )
      document.forms[0].action = '?go = 提交';  //表单提交标记
      document.forms[0].submit();
}
function checkAll(touch,name)
{
    var el = document.getElementsByTagName('input');     //读取表单内所有元素
```

```
    var len = el.length;                        //读取元素的个数
    for(var i = 0; i < len; i++)                //循环元素个数的次数
    {
        if((el[i].type == "checkbox") && (el[i].name == name))  //读取所有名为 name 的复选框
        {
            if(document.getElementById(touch).checked)  //若用户单击“全选”
            {
                el[i].checked = true;           //选中所有复选框
            }
            else{                               //用户取消全选
                el[i].checked = false;          //取消所有复选框被选中
            }
        }
    }
}
 function chgFace(){
  document.forms[0].submit();
}
</SCRIPT>
</HEAD>
<%
uid = Trim(Request("uid"))
pwd = Trim(Request("pwd"))
sex = Trim(Request("sex"))
userFace = Trim(Request("userFace"))
email = Trim(Request("email"))
character = Trim(Request("character"))
interest = Trim(Request("interest"))
go = Trim(Request("go"))
If go <>"" Then
 %>
提交成功!
<%
End If
 %>
```

注册页面如图 15-21 所示。

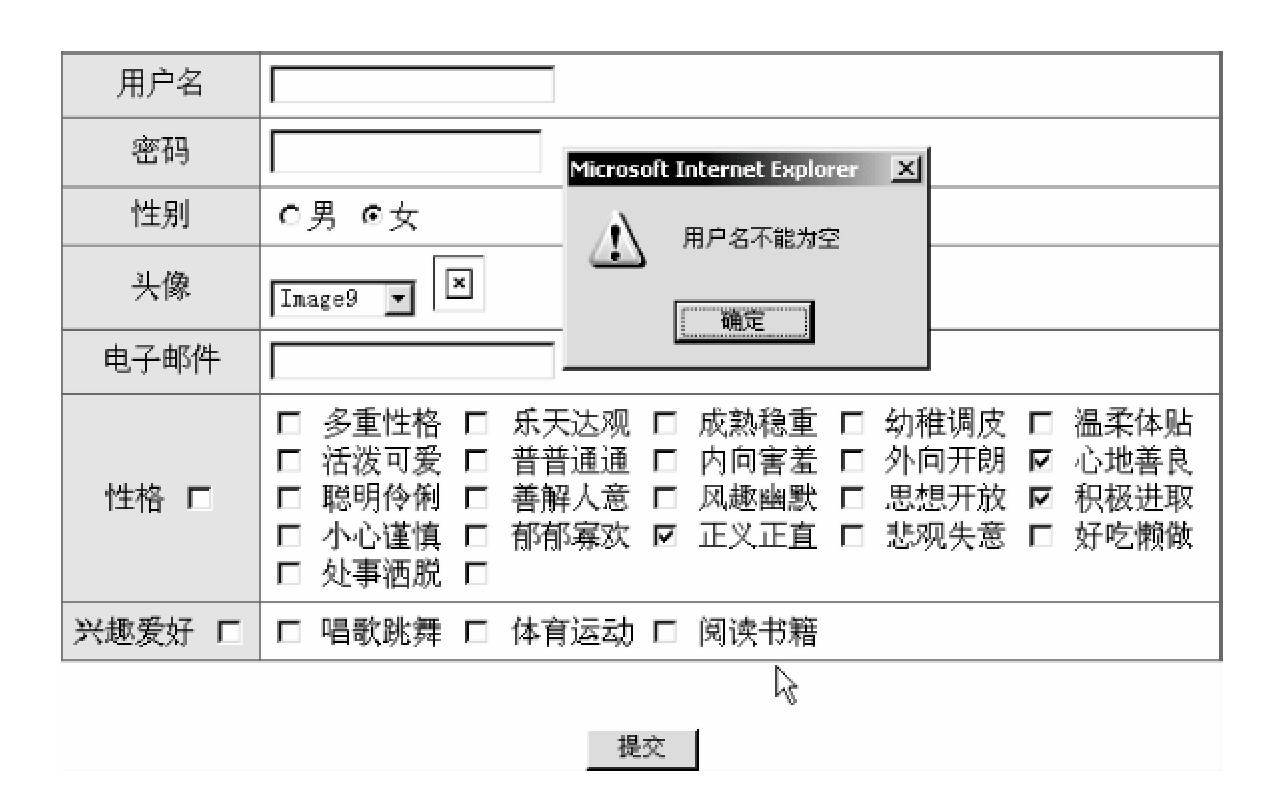


图 15-21　注册页面的验证

五、操作技巧与注意事项

(1) 由于一周有 7 天,每 7 天是一个循环,因此用 if i mod 7=0 then 语句来判断是否已经是一周的第七天,即星期六,如果是,则动态地重新输出一行,然后把之后的数字输入到新产生的表格中,一直循环,直到 today 的值大于表示本月最大日期的值。

(2) 注册页面中,采用循环语句判断是否选中复选框。

(3) 注意文本框中验证用户名是否为空的方法。

六、思考题与练习

(1) 利用 JavaScript 上传图像的方法。

(2) 利用 VBScript 动态输出九九乘法表的方法。

15.4 实训 4——留言板的实验

一、实验类别

操作型实验。

二、实验目的

(1) 理解数据库访问技术 ODBC。

(2) 熟练掌握在 ASP 中连接 Access 数据库的方法。

三、实验内容及要求

(1) 掌握 IIS 信息服务器的安装和启动。

(2) 使用 ODBC 数据访问接口访问 Access 数据库。创建一个 ODBC 数据源,在 ASP 应用程序中直接使用该数据源访问 Access 数据库中的数据。

(3) 能够在留言板上发言,并将发言信息写入数据库中。

(4) 能够查看留言信息,管理员可以回复留言信息。

四、实验方法及步骤

1. IIS 信息服务器的安装和启动

1) IIS 的安装

选择"开始"→"设置"→"控制面板"→"添加/删除程序"命令,在"添加或删除程序"对话框中单击"添加/删除 Windows 组件"按钮,就会弹出 "Windows 组件向导"对话框。在其中选择"Internet 信息服务(IIS)",然后单击"下一步"按钮,随后根据提示一步步安装即可。

2) IIS 的启动

在"Internet 信息服务"对话框左侧的树窗格中右击"本地计算机"图标,从弹出的快捷菜单中选择"所有任务"→"重新启动 IIS"命令。要浏览 ASP 页面,只须右击 ASP 页面,单

击“浏览”按钮，即可运行。

2. 创建 ODBC 数据源

（1）打开控制面板，双击管理工具，如图 15-22 所示。



图 15-22　管理工具界面

（2）双击“数据源(ODBC)”图标，启动“ODBC 数据源管理器”，如图 15-23 所示。

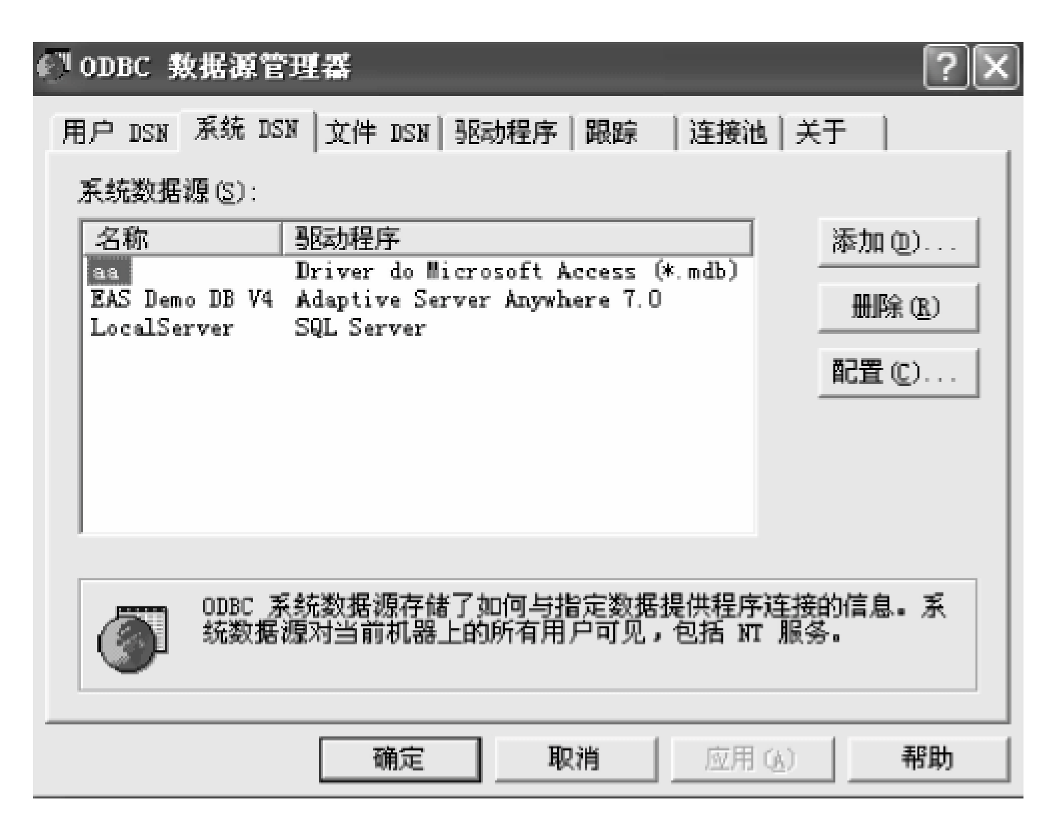


图 15-23　ODBC 数据源管理器

（3）切换到“系统 DSN”标签，单击“添加”按钮，在已安装的驱动程序列表中为数据源选择一个数据库驱动程序（如 Access），单击“完成”按钮，如图 15-24 所示。



图 15-24　为数据源选择一个数据库驱动程序

(4) 在 ODBC 设置界面中，如图 15-25 所示，从指定的数据源名中选择已经建立好的数据库名(如学生管理系统. mdb)，然后单击“确定”按钮，完成数据库源的选择。



图 15-25 指定数据源名并选择数据库服务器

3. 打开 Access 数据库，创建数据库表

(1) 进入 Access，选择“文件”→“新建空数据库”→“文件名：db. mdb”→“使用设计器创建表”。

(2) 创建表，表名如下所示。

- admin 表：主要负责建立用户的名称和密码的登录信息，如图 15-26 所示。
- gb 表：主要负责建立留言信息的内容，如图 15-27 所示。



图 15-26 admin 表



图 15-27 gb 表

4. 创建登录界面文件 login.asp

建立网络留言本首页，如图 15-28 所示，操作方法如下：

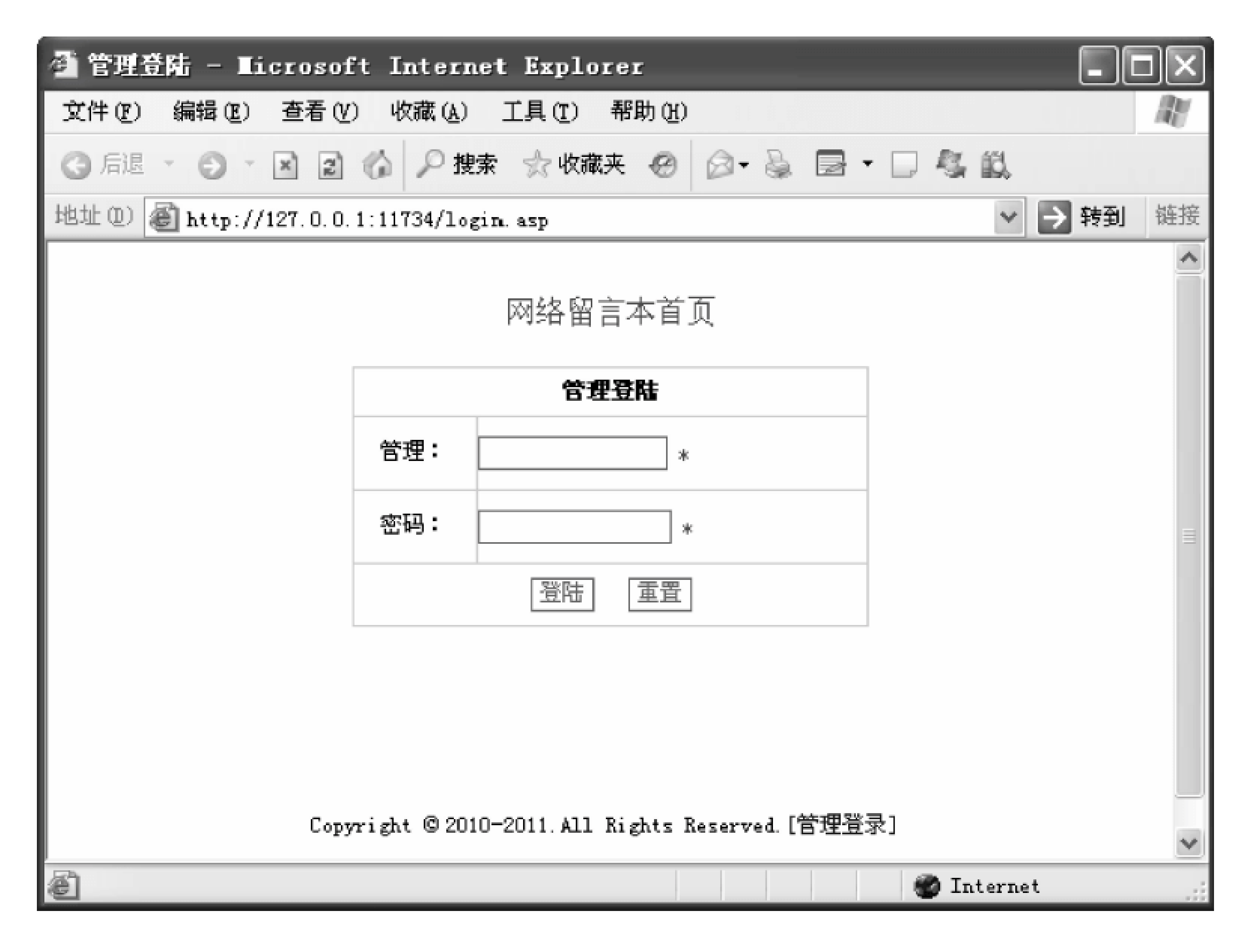


图 15-28　网络留言本 login.asp

（1）打开 Dreamweaver 软件，建立一个 7 行 3 列的表格。表格内容设置如表 15-1 所示。

表 15-1　登录表的对象

对　象　名	属　性　名	属　性　值
Label1	text	姓名
	id	空
Label2	text	密码
	id	空
Text1	id	txtUserName
	text	空值
Text2	id	txtUserPassWord
	type	password
	text	空值
Button1	text	登录
	id	Submit
Button2	text	重置
	id	Reset

（2）建立数据库连接文件 conn.asp。

在记事本中写入如下代码，命令为 conn.asp 并保存。

【代码】

```
<%

Dim db,conn,connstr        '定义 3 个变量名,分别是 db、conn、connstr
db = "_data/#data_cnpn.mdb"    '建立数据库名字
set conn = Server.CreateObject("ADODB.Connection")'创建 ADO DB 连接
connstr = "Provider = Microsoft.Jet.OLEDB.4.0;Data Source = " & Server.MapPath(""&db&"")
'设定驱动服务器的提供者和服务器的路径
conn.Open connstr      '打开连接
%>
<%
'设定每页显示留言数量
Dim listnum
listnum = 10
'过滤 HTML 代码
function HTMLEncode(fString)
fString = replace(fString, ">", "&gt;")
fString = replace(fString, "<", "&lt;")
fString = Replace(fString, CHR(32), " ")
fString = Replace(fString, CHR(34), "&quot;")
fString = Replace(fString, CHR(39), "&#39;")
fString = Replace(fString, CHR(13), "")
fString = Replace(fString, CHR(10) & CHR(10), "</P><P> ")
fString = Replace(fString, CHR(10), "<BR> ")
HTMLEncode = fString
end function
'检查输入的 E-mail 是否合法
function IsValidEmail(email)
Dim names, name, i ,c
IsValidEmail = true
names = Split(email, "@")   '使用拆分函数,校验邮箱
if UBound(names) <> 1 then
   IsValidEmail = false
   exit function
end if
for each name in names
   if Len(name) <= 0 then
     IsValidEmail = false
     exit function
   end if
   for i = 1 to Len(name)
     c = Lcase(Mid(name, i, 1))
     if InStr("abcdefghijklmnopqrstuvwxyz_-.", c) <= 0 and not IsNumeric(c) then
       IsValidEmail = false
       exit function
     end if
   next
   if Left(name, 1) = "." or Right(name, 1) = "." then
      IsValidEmail = false
      exit function
```

```
    end if
next
if InStr(names(1), ".") <=  0 then
    IsValidEmail  =  false
    exit function
end if
i  =  Len(names(1)) - InStrRev(names(1), ".")
if i <> 2 and i <> 3 then
    IsValidEmail  =  false
    exit function
end if
if InStr(email, "..") > 0 then
    IsValidEmail  =  false
end if
end function
```

5. 添加留言界面文件 add.htm

打开 Dreamweaver，创建 12 行 3 列的表格，表格界面如图 15-29 所示。表格对象设置如表 15-2 所示。

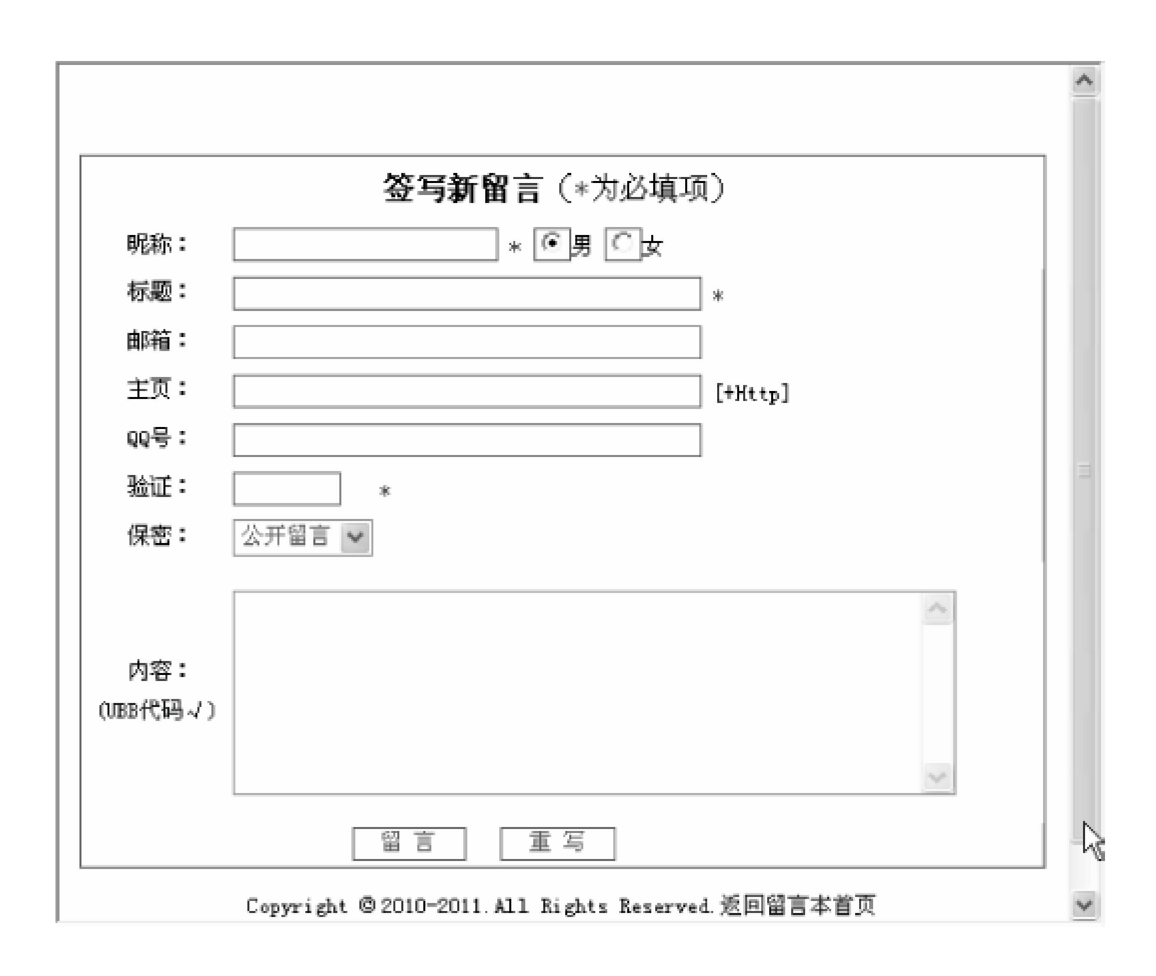


图 15-29　填写新留言

表 15-2　留言表对象

对　象　名	属　性　名	属　性　值
Label1	text id	昵称 空
Label2	text id	标题 空
Label3	text id	邮箱 空
Label4	text id	主页 空

续表

对　象　名	属　性　名	属　性　值
Label5	text	QQ号
	id	空
Label6	text	验证
	id	空
Label7	text	内容
	id	空
Text1	Id	txtUserName
	text	空值
Text2	id	txtUserPassWord
	type	password
	text	空值
Button1	text	登录
	id	Submit
Button2	text	重写
	id	Reset

五、操作技巧与注意事项

(1) 设置数据源时,注意 ODBC 数据源的设置方法。
(2) 制作主页界面,要求在主页页面中引入 conn.asp 文件,引入方法如下:

```
<! -- #include file = "inc/conn.asp" -- >
```

六、思考题与练习

(1) 简述留言板的原理。
(2) 讨论留言板的数据库设计。
(3) 制作留言板的登录界面。
(4) 制作留言板的留言界面。

15.5 实训5——购物车的实验

一、实验类别

操作型实验。

二、实验目的

(1) 理解购物车的购物流程。
(2) 了解 B2C 的电子商务购物模式。

（3）掌握购物车的展示、添加、购物功能。

三、实验内容及要求

本次实验主要以购物车的购买流程为主要内容，重点掌握以下5个方面：

（1）商品列表展示。

（2）商品详细信息展示。

（3）向购物车中添加商品。

（4）查看购物车。

（5）收银台结算。

四、实验方法及步骤

1．购物功能分析

1）系统模块划分

系统模块划分为商品展示模块、购物车模块、收银台结算模块。

商品展示模块包括商品列表、商品详细信息，如图15-30所示。

购物车模块包括向购物车中添加商品、查看购物车、修改已购买商品的数量、移除购物车中指定商品，如图15-31所示。

收银台结算模块包括填写用户信息、显示用户信息，如图15-32所示。

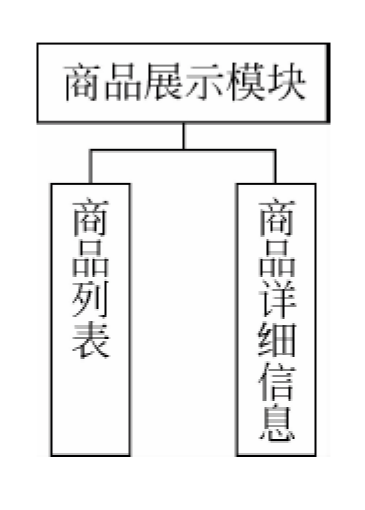


图15-30　商品展示模块

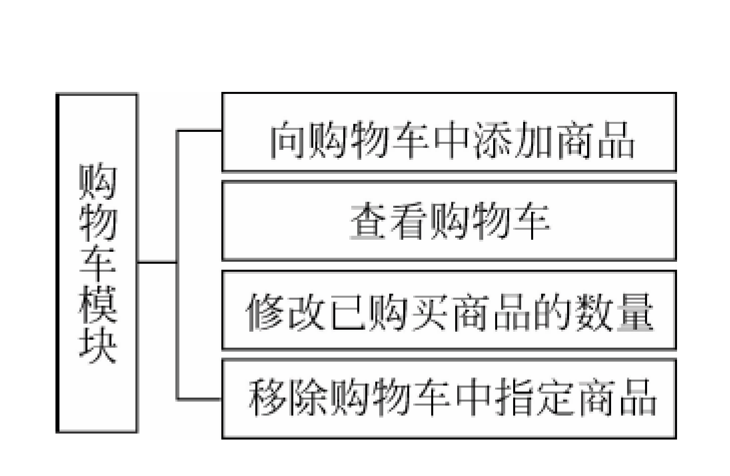


图15-31　购物车模块

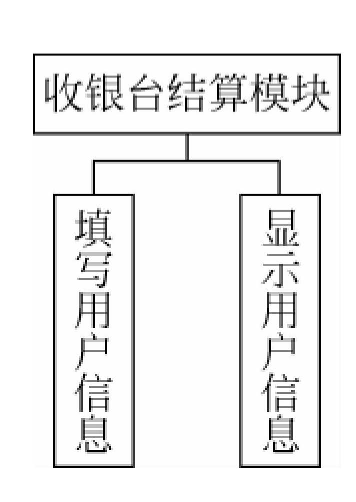


图15-32　收银台结算模块

2）购物的过程

首先登录系统，进入index.asp页面浏览商品列表，看到自己喜欢的商品后，单击图片，查看商品详细信息的detail.asp页面。确定购买该商品，把它放入购物车cart.asp中。在购物车中可存放多个商品，这些商品都是自己想要购买的商品，所以在网站购物中可同时购买多个商品。确定购买的商品后，查看购物车的内容，进入收银台进行结算的account.asp页面，系统会自动为每个商品分配订单编号，完成支付功能。购物成功后，退出系统。购物流程展示如图15-33所示。

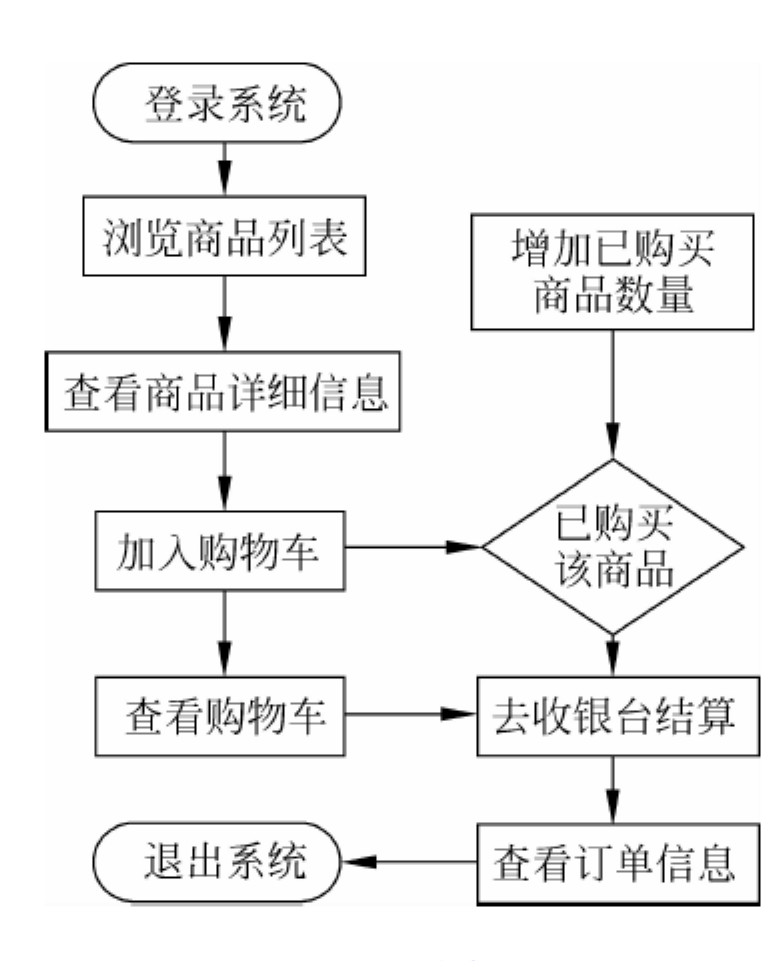


图15-33　购物流程

2．商品展示模块

将商品以静态页的形式放置到网页中，并为每个商

品添加固定的链接，便于区分。

操作思路：

（1）创建 index.asp 文件（如图 15-34 所示），手动维护 3 条商品记录。



图 15-34 index.asp 购物车首页

（2）创建 detail.asp 文件（如图 15-35 所示），接收从首页（index.asp）传递过来的参数，根据商品标识不同，显示各自对应的商品图片。



图 15-35 detail.asp 详细页面

（3）在 detail.asp 中将商品名称、价格显示在页面中。

（4）在页面中添加购物车链接，将商品信息通过链接传递给购物车模块。

关键代码：

在 index.asp 中实现进入 detail.asp 的代码：

```
<a  href = "detail.asp?proId = 1&proName = &#24800; &price = 2499"  target = _blank >
```

在 detail.asp 中，从图像控件中选择用户需要商品的详细信息代码：

```
<%
Select Case proId
  Case "1" proPic = "1.jpg"
  Case "2" proPic = "2.jpg"
  Case "3" proPic = "3.jpg"
End Select
%>
```

3. 购物车模块

用户在商品详细信息页(detail.asp)单击"购买"按钮后,进入购物车页面(cart.asp,如图 15-36 所示),该页面使用一个表格存放用户选择过的所有商品信息。

图 15-36　购物车页面(cart.asp)

操作步骤如下:

(1) 单击"购买"某商品后,进入购物车页面(cart.asp),首先读取 Cookie 中的字符串,在 Cookie 字符串中查找新提交的商品。

(2) 通过对变量 position 的判断,确定该商品是否已被购买过,如果变量的值不为 0,则只修改购买数量,不添加新的记录;如果变量的值为 0,重新添加购买商品。

(3) 将 Cookie 中的商品信息显示在购物车页面(cart.asp)中。

关键代码:

(1) 在 cart.asp 中,读取 Cookie 中的字符串代码:

```
<%
proId = Trim(Request("proId"))                  '获取编号,并赋值给变量 proID
proName = Trim(Request("proName"))              '获取名称,并赋值给变量 proName

proIdString = Request.Cookies("proIdString")  '分别获取 Cookie 中的字符串
proNameString = Request.Cookies("proNameString")
priceString = Request.Cookies("priceString")
numberString = Request.Cookies("numberString")
position = InStr(proIdString&"^","^"&proId&"^")
```

(2) 确定所需商品是否购买的代码:

```
If position <> 0 Then              '若变量的值不为 0,修改购买数量,不添加新的记录
  arrNM = Split(numberString,"^")
  numberString = ""
  For i = 1 To UBound(arrNM)     '从第一个商品开始循环
    If i = (position + 1)/2 Then
      newNumber = CInt(arrNM(i)) + CInt(number)
    Else
```

```
        newNumber = arrNM(i)
      End If
      numberString = numberString&"^"&newNumber
    Next
  Else
  End If
```

4. 收银台结算

用户填写联系方式、送货地址等信息，完成购物过程(account.asp)，如图 15-37 所示。



图 15-37　结算页面(account.asp)

操作步骤如下：

(1) 在页面(account.asp)中编写判断函数，保证用户信息的完整性。

(2) 表单(account.asp)提交后验证表单项是否为空。

(3) 表单(account.asp)验证通过后(验证内容有用户名正确、密码正确、购物车有物品、支付成功)，向用户提示购买成功，同时清空购物车。

关键代码：

在 account.asp 中用户信息得到确认后，调用子过程，清空 Cookie 中的购物车信息代码：

```
If go <> "" Then
  Call checkNull(name,"姓名")              '检查表单项是否为空
  Call checkNull(address,"地址")
  Call checkNull(contract,"联系方式")
  Call checkNull(order,"收货方式")

Call setCookie("proIdString","")          '清空购物车原有内容
  Call setCookie("proNameString","")
  Call setCookie("priceString","")
  Call setCookie("numberString","")
%>
```

五、操作技巧与注意事项

(1) 从表单获取数据和发送数据时，注意使用 Response 对象和 Request 对象。
(2) 使用 Select…Case 语法可以判断用户需要的商品信息。
(3) 利用 Cookie 对象保存用户信息，并且能够读取 Cookie 中保存的信息。
(4) 添加新记录时，使用字符串拼接及拆分的方法。
(5) 在用户信息得到确认后，调用子过程清空 Cookie 中的购物车信息。

六、思考题与练习

(1) 简述电子商务的几种模式。
(2) 简述购物车的购买流程。
(3) 制作购物车展示模块。
(4) 制作购物车模块。
(5) 制作收银台结算模块。

15.6 实训 6——SQL Server 数据库与基本表的创建和管理

一、实验类别

验证型实验。

二、实验目的

使学生熟练掌握和使用 SQL 或 Transact-SQL、SQL Server 企业管理器创建数据库、表、索引和修改表结构，学会使用 SQL Server 查询分析器接收 Transact-SQL 语句和进行结果分析。

三、实验内容及要求

(1) 创建数据库和查看数据库属性。
(2) 创建表，确定表的主键和约束条件，为主键建立索引；查看和修改表结构。
(3) 熟悉 SQL Server 企业管理器和查询分析器工具的使用方法。
(4) 建立 Student 数据库及其数据表(学生表 S、课程表 C 和选课表 SC)和表间联系。
(5) 选择合适的数据类型；定义必要的索引、列级约束和表级约束。

四、实验步骤及方法

1. 使用企业管理器创建 Student 数据库的步骤

(1) 选择“开始”→“程序”→Microsoft SQL Server→“企业管理器”，打开企业管理器。
(2) 选中需要在其上创建数据库的服务器，单击前面的“+”号，使其展开为树形目录。

(3) 选中“数据库”文件夹，右击鼠标，从弹出的快捷菜单中选择“新建数据库”命令，在出现的“数据库属性”对话框的“常规”选项卡中输入数据库名 Study，如图 15-38 所示。



图 15-38 “数据库属性”对话框

(4) 单击“数据文件”标签，输入数据库文件的属性，包括文件名、存放位置和初始大小等，如图 15-39 所示。在选择文件位置时，可以单击“位置”列的 ... 按钮，在调出的文件选择器中进行位置选择。文件属性参数说明如下。



图 15-39 “数据库属性”对话框中的“数据文件”选项卡

① “文件自动增长”复选框：选中后允许文件装满数据时自动增长。

② “文件增长”选项组：允许文件自动增长时，设置每次文件增长的大小。选择“按兆字节”选项，则设置文件增长为固定的大小，单位是 MB；选择“按百分比”选项，则文件按指

定比例数增长，单位是％。

③“最大文件大小”选项组：设置当允许文件扩展时数据文件能够增长的最大值。选择“文件增长不受限制”选项，可使文件无限增长，直到用完磁盘空间；选择“将文件增长限制为”选项，要设置文件最多达到的固定值。

(5) 单击“事务日志”标签，输入日志文件属性，包括文件名、存放位置、大小和文件属性。单击“确定”按钮，关闭对话框。在企业管理器窗口中出现 Student 数据库标志，这表明建库工作已经完成。

2. 查看和设置数据库属性

右击 Student 数据库标志，在弹出的快捷菜单中选择“属性”选项，出现“Study 属性”对话框，它由 6 个选项卡构成。其中，“文件组”选项卡用来设置数据库的文件组；“权限”选项卡用来设置用户对该数据库的访问权限；“选项”选项卡如图 15-40 所示，用来设置数据库的操作许可、故障还原模型以及数据库在进行操作时的行为特征。



图 15-40　“Study 属性”对话框中的“选项”选项卡

3. 删除数据库

右击要删除的数据库，从弹出的快捷菜单中选择“删除”命令，再单击确认删除对话框中的“确认”按钮。

4. 新建表

在 SQL Server 2000 的数据库中，文件夹是按数据库对象的类型建立的，文件夹名是该数据库对象名。在企业管理器中，双击已定义好的 Student 数据库后，会看到它自动设置了关系图、表、视图、存储过程、用户、角色、规则、默认等文件夹。

（1）新建表的方法是右击 Student 数据库中的表文件夹，从弹出的快捷菜单中选择“新建表”命令，出现如图 15-41 所示的输入表结构对话框。



图 15-41 输入表结构对话框

（2）标识用于设置字段名、初始值以及步长，以便让 SQL Server 2000 自动填写该列的值。一个表只允许有一列具有标识，其数据类型只能为 int、smallint、tinyint、decimal(P,0)或 numeric(P,0)，不允许为空值。

在输入表结构对话框中右击，则会弹出菜单。选择“设置主键”选项，则定义当前字段为主键，表中第一列处会显示钥匙图案；选择“插入行”选项，则在当前字段处插入一个新行；选择“删除列”选项，则删除当前字段；选择“属性”选项，可弹出“属性”对话框，如图 15-42 所示，可以在其中定义索引/键、与其他表间的关联和约束等属性。



图 15-42 “属性”对话框

字段输入完后，就可以关闭建表对话框了。在弹出的对话框中输入表名，单击“确定”按钮后，表结构的建立就完成了。

5. 定义表的完整性约束和索引

1）定义索引和键

(1) 查看、修改或删除索引时，先要在图 15-42 中的“选定的索引”下拉列表框中选择索引名，其索引内容就显示在表中。也可以直接在表中修改索引内容，如改变索引列名、排序方式等。对于不需要的索引可以单击“删除”按钮直接删除。

(2) 新建一个索引时，单击“新建”按钮，并在下面的表中输入索引名、索引列名及排序方式。

(3) 设置“创建 UNIQUE”复选框，确定是否为唯一索引约束。设置“创建为 CLUSTERED”复选框，确定是否为聚簇索引。

2）定义表间关联

(1) 单击“关系”标签，如图 15-43 所示。查看、修改或删除表关联时，先要在“选定的关系”下拉列表框中选择关系名，其关联内容就显示在表中，也可以直接在表中修改关联内容，如改变主键、改变外键等。对于不需要的关联可以单击“删除”按钮直接删除。



图 15-43　“属性”对话框中的“关系”选项卡

(2) 新建一个关联时，单击“新建”按钮，选择库中的关联表后，在表中输入关联名、主键和外键。

(3) 设置“创建中检查现存数据”复选框，确定新建关联时是否对数据进行检查，要求符合外键约束；设置“对复制强制关系”复选框，确定在进行数据复制时是否要符合外键约束；设置“对 INSERT 和 UPDATE 强制关系”复选框，确认在进行数据的插入和更新时是否要符合外键约束；设置“级联更新相关的字段”复选框和“级联删除相关的记录”复选框，确认被参照关系的主键值被修改时是否也将参照表中的对应的外码值修改，而被参照关系的主

键被删除时，是否也将参照表中对应外键的记录删除。

3）定义 CHECK 约束

（1）单击“CHECK 约束”标签，如图 15-44 所示。查看、修改或删除 CHECK 约束时，要在“选定的约束”下拉列表框中选择约束名，其约束内容就显示在约束表达式框中，也可以直接在框中修改约束表达式。对于不需要的约束可以单击“删除”按钮直接删除。



图 15-44 “属性”对话框中的“CHECK 约束”选项卡

（2）新建一个 CHECK 约束时，单击“新建”按钮，输入约束名和约束表达式。在此设置 S 表中 Sex 属性的取值范围是“男”或者“女”；Sage 属性的取值范围为 15～45 之间以及 C 表和 SC 表中相应属性的约束。

（3）设置“创建中检查现存数据”复选框，确认在创建约束时是否对数据进行检查，要求符合约束要求；设置“对复制强制约束”复选框，确认在进行数据复制时是否要符合约束条件；设置“对 INSERT 和 UPDATE 强制约束”复选框，确认在进行数据的插入和更新时是否要求符合约束条件。

6. 修改表结构

（1）使用企业管理器。首先选中该表，右击鼠标，从弹出的快捷菜单中选择“设计表”命令，则出现输入表结构对话框，在此用户对原有内容进行修改。

（2）使用 SQL 语句。在企业管理器中选择“工具”→“SQL 查询分析器”菜单项，打开查询分析器。在此输入修改表结构的命令 ALTER TABLE，有 3 种修改方式：ADD 方式，用于增加新的字段名；ALTER 方式，用于修改某些列；DROP 方式，用于删除完整性约束。

五、操作技巧与注意事项

（1）数据库的名称最长为 128 个字符，且第一个字符必须是字母字符，不区分大小写。

（2）一个服务器在理论上可以管理 32 767 个数据库。一个数据库至少应包含一个数据

文件即主数据文件和一个事务日志文件。

(3) 列名也称为字段名，它类似于变量名，其命名规则与变量一致。列名中不允许出现空格，一张表也不允许有重复的列名。

(4) 数据类型列中的数据类型是通过数据类型下拉框选择的，而不是直接输入数据类型字符输入的。

(5) 每一个表只能创建一个聚簇索引，可以创建多个(249 个)非聚簇索引。

(6) 正在使用的数据库不能删除；系统表不能删除。

六、思考题与练习

(1) 为什么要建立索引？索引越多越好，这句话是否正确？聚簇索引与非聚簇索引各有什么特点？

(2) Student 数据库的 SC 表中，学号、课程号采用数值型，还是采用字符型？采用哪种数据类型更好？

(3) 在 SQL 中列级数据完整性约束有哪几类？

(4) 如何设置默认约束？将 S 表中 Sdept 属性的默认值设置为“计算机系”。

15.7　实训 7——数据查询实验

一、实验类别

验证型实验。

二、实验目的

使学生掌握 SQL Server 查询分析器的使用方法，加深对 SQL 和 Transact-SQL 语言的查询语句的理解与应用。熟练掌握简单表的数据查询和数据连接查询的操作方法。

三、实验内容及要求

(1) 简单查询操作，包括投影、选择条件表达、数据排序等。

(2) 连接查询操作，包括等值连接、自然连接、一般连接、外连接、左连接、右连接和自身连接等。

四、实验方法及步骤

(1) 将查询需求用 Transact-SQL 语言表示。

(2) 在 SQL Server 查询分析器的输入区中输入 SQL 或 Transact-SQL 查询语句。

(3) 设置查询分析器结果区为 Standard Execute(标准方式)或 Execute to Grid 方式。

(4) 发布执行命令，查看查询结果；如果结果不正确，进行修改，直到正确为止。

单击“开始”菜单→Microsoft SQL Server →“企业管理器”，打开企业管理器，先选中

Student 数据库,再选择“工具”→“SQL 查询分析器”菜单项;或者选择“开始”菜单中的 Microsoft SQL Server→“查询分析器”,均可打开查询分析器,如图 15-45 所示。查询分析器的主要作用是编辑 Transact-SQL 语句,将其发送到服务器,并将执行结果及分析显示出来(或进行存储)。查询分析功能主要通过测试查询成本、判断该查询是否需要增加索引以提高查询速度,并可以实现自动建立索引的功能。在查询生成器的左边窗口是对象浏览器,其中以树形结构列出数据库对象;右上方是输入 SQL 代码区域,用于输入 SQL 的查询语句;右下方为结果区,用于显示查询结果和分析结果。窗口的下半部分是 SQL 语句执行结果的信息窗口。其中,工具栏上的“▶”图标表示执行查询,“√”图标表示分析查询,“■”图标表示取消执行查询。

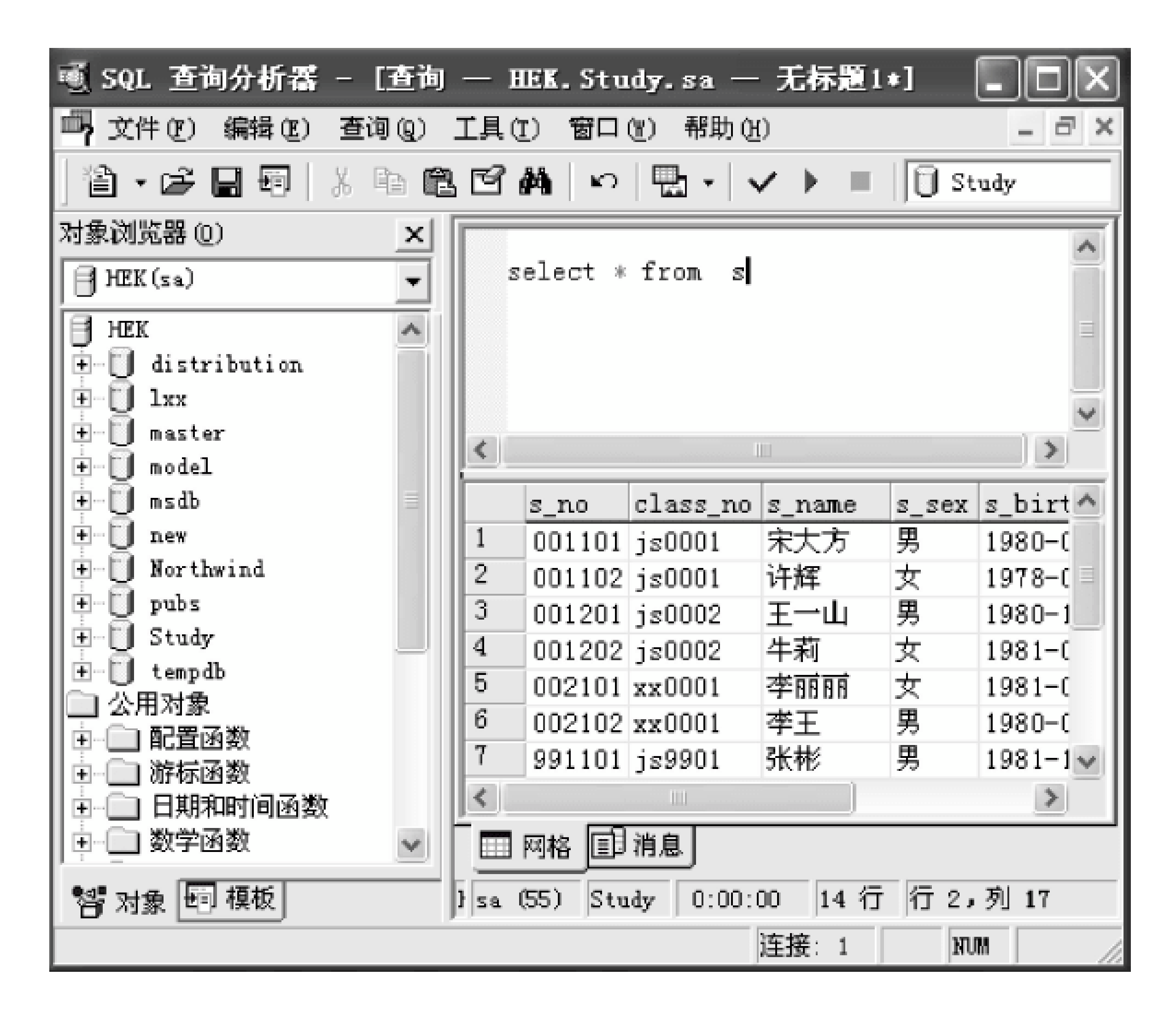


图 15-45 查询分析器界面

用 Transact-SQL 语句进行以下查询操作。

1) 简单查询

(1) 查询全体学生的学号和姓名。

(2) 查询计算机系的所有学生信息。

(3) 查询选修了课程的学生学号。

(4) 查询选修了 C02 课程的学号和成绩,并对查询结果按成绩降序排列,如果成绩相同则按学号升序排列。

(5) 查询选修了课程 C04 且成绩在 80～90(包括 80 和 90)分之间的学生学号和成绩,并将成绩乘以 0.9 系数输出。

(6) 查询全体学生的姓名及其出生年份。

(7) 查询缺少成绩的学生学号和课程号。

(8) 查询名字中第二个字为“小”或“大”字的学生的姓名和学号。

2) 连接查询

(1) 查询每个学生的信息及其选课的信息。

(2) 查询计算机系学生的选课情况,要求列出其学号、姓名、选修的课程号及成绩。

(3) 查询与刘晨在同一个系学习的学生的姓名和所在系。

(4) 查询信息系选修 VB 课程的学生的学号、姓名、成绩。

(5) 查询选修了数据库基础课程且成绩在 85 分以上的学生学号、姓名及成绩。

(6) 用 Transact-SQL 语句表示学生表 S 和选修课程表 SC 之间的左外连接、右外连接和完全外连接。

五、操作技巧与注意事项

(1) 如果想去掉查询结果表中的重复行,必须有 distinct 关键字。

(2) 查询结果的几种处理方式。

(3) 输入 SQL 语句时,应注意语句中均使用西文操作符号。

(4) 通过 top 关键字来限制返回到结果集中的行数,其基本格式如下:

```
Select  top  n  [ percent ]  *  from  表名
```

如果没有 percent 关键字,n 就是返回前几条记录的行数;如果带有 percent 关键字,n 则是返回结果集中行数的百分比。

(5) 对数据类型为 text 和 image 的字段不能使用 Order by 进行排序;在 Order by 子句中也不能使用子查询、聚合函数和常量表达式。

六、思考题与练习

(1) 在查询某一表中所有记录时所使用的特定符号是________。

(2) 在 Select 语句中使用________关键字可消除字段的重复项。

(3) 如果要查询 S 表中前 4 行记录,应该使用的语句是________。

(4) 有一种查询叫模糊查询,在 SQL 语句中要使用________关键字来实现这种查询方式。

(5) 如何提高数据查询和连接速度?

(6) 理解内连接、左外连接和右外连接的含义及表达方法。

15.8 实训 8——数据的导入/导出与数据库备份/还原

一、实验类别

验证型实验。

二、实验目的

(1) 使学生掌握 SQL Server 中利用 DTS 向导实现数据的导入/导出。

(2) 使学生了解 SQL Server 的数据备份和还原机制,熟练掌握使用企业管理器进行数据库的备份/还原方法。

三、实验内容及要求

(1) 利用 SQL Server 2000 中提供的 DTS 向导实现数据的导出。
(2) 利用 SQL Server 2000 中提供的 DTS 向导实现数据的导入。
(3) 使用企业管理器创建一个备份 Student 数据库,要求每星期对数据库备份一次。
(4) 在企业管理器中还原 Student 数据库。

四、实验方法及步骤

1. 利用 SQL Server 2000 中提供的 DTS 向导实现数据的导出

将 Student 数据库中的 S 表导出,格式为 Execl 电子表格。

打开企业管理器,选择菜单"工具"→"向导",在弹出的窗口中单击"数据转换服务"左边的"+"号,使之展开,选择"DTS 导出向导"选项,单击"确定"按钮。或者右击 SQL Server 2000 企业管理器的控制树的"数据转换服务"节点,选择"所有任务",单击"导出数据(E)"。以上方法均可弹出数据的导出向导对话框。单击"下一步"按钮,通过以下 6 个步骤即可完成数据导出:

(1) 选择数据源的类型、认证方式、将要导出数据的数据库(导出数据时,活动连接是默认的源服务器),单击"下一步"按钮。

(2) 选择将数据复制到何处。在此选择 Microsoft Excel 97、Excel 2000,并指定文件名及路径,单击"下一步"按钮。

(3) 选择从源数据库得到目标数据的方式(三选一,在此选从源数据库复制表),单击"下一步"按钮。

(4) 选择数据表(至少选一个表,也可以选多个表),单击"下一步"按钮。

(5) 选择传输数据的执行方式以及是否保存 DTS 包,单击"下一步"按钮。

(6) 系统显示以上设置信息摘要,单击"完成"按钮即可。

2. 利用 SQL Server 2000 中提供的 DTS 向导实现数据的导入

首先在 Microsoft Access 中创建 Student.mdb 数据库以及创建 S 表(sno, sname, birthday),向 S 表中插入一条记录。然后打开企业管理器,选择菜单"工具"→"向导",在弹出的窗口中单击"数据转换服务"左边的"+"号,使之展开,选择"DTS 导入向导"选项,单击"确定"按钮。或者右击 SQL Server 2000 企业管理器的控制树的"数据转换服务"节点,选择"所有任务",单击"导入数据(T)"。以上两种方法均可弹出数据的导入向导对话框。单击"下一步"按钮,通过以下 6 个步骤即可完成数据导入:

(1) 选择数据源的类型(在此选 Microsoft Access),输入将要导入数据的数据库文件名(导入数据时,活动的服务器连接的就是目标服务器),单击"下一步"按钮。

(2) 选择将数据复制到何处。在此选择当前默认连接的 Student 数据库,并指定认证方式、用户名和密码,单击"下一步"按钮。

(3) 选择从源数据库得到目标数据的方式(三选一,在此选从源数据库复制表),单击"下一步"按钮。

（4）选择数据表（至少选一个表，也可以选多个表），单击“下一步”按钮。

（5）选择传输数据的执行方式以及是否保存 DTS 包，单击“下一步”按钮。

（6）系统显示以上设置信息摘要，单击“完成”按钮即可。

3. 使用企业管理器创建备份文件

（1）右击“管理”文件夹下的“备份”节点，从弹出的快捷菜单中选择“新建设备”选项。

（2）在出现的对话框中输入备份文件的逻辑名称，确定备份文件的文件名（可通过 ... 按钮改变备份文件的默认磁盘文件路径和文件名），单击“确定”按钮。

4. 备份 Student 数据库

（1）在企业管理器中右击备份的 Student 数据库，从弹出的快捷菜单中选择“所有任务”中的“备份数据库”选项，则出现数据库备份对话框。对话框中有“常规”和“选项”两个页面。

（2）“常规”选项卡如图 15-46 所示。第一区域是选择要备份数据库；第二区域是选择备份的类型（完全、差异备份等）；第三区域是指定备份介质，可通过“添加”按钮选择备份的设备或文件，通过“内容”按钮查看备份设备或文件中的内容，通过“删除”按钮删除已经选定的备份设备文件；第四区域是选择将备份保存到备份文件时的覆盖模式；最后一个区域是选择执行备份的日期和时间，可通过 ... 按钮设定备份计划需要执行的操作。

（3）“选项”选项卡如图 15-47 所示，设置是否进行备份文件验证，是否初始化备份文件。

单击“确定”按钮之后，出现数据库备份成功信息框，单击“确定”按钮。



图 15-46 数据备份的“常规”选项卡



图 15-47 数据备份的“选项”选项卡

5. 还原 Student 数据库

（1）在企业管理器中右击要还原的数据库 Student，从弹出的快捷菜单中选择“所有任

务”中的“还原数据库”选项，则出现数据库还原对话框。该对话框中有“常规”和“选项”两个选项卡。

(2) 在“常规”选项卡中选择数据库以及还原方式。不同的选项，其选项页面和设置恢复的方法也不同。

(3) 在“选项”选项卡中设置附加特征。

(4) 当完成了“常规”和“选项”选项卡的参数设置后，单击下面的“确定”按钮，SQL Server 就开始进行数据库还原操作，屏幕上显示出还原进度的对话框。

五、操作技巧与注意事项

(1) 注意：只有输入正确的所有者口令才可以修改该 DTS 包的定义和执行 DTS 包。若只输入用户口令，只能执行 DTS，不能修改 DTS 包定义。

(2) 在备份设备属性对话框中，磁带驱动器名单选按钮是灰色的，不能使用。这是因为本服务器上没有安装磁带驱动器。只有安装了磁带驱动器，才可以使用该选项。

(3) 使用备份向导不能选择执行文件或文件组备份类型。

(4) 使用 Backup Database 语句、企业管理器和向导 3 种方法都可以完成数据库的备份操作，但是这些命令和工具应用的范围是不同的。一般地，可以使用企业管理器工具执行数据库的备份操作，它提供了大多数常用的数据库备份选项。Backup Database 语句是最完整的工具，适合于 DBA 使用。对于初学者，最好使用向导执行数据库的备份操作。

六、思考题

(1) 为什么要执行数据库的备份操作？

(2) 何时执行数据库的还原操作？

(3) 数据的导入和导出是如何操作的？

15.9 实训 9——ASP 连接 SQL Server 数据库

一、实验类别

操作型实验。

二、实验目的

(1) 了解 ADO 技术的原理和方法。

(2) 掌握 connection、command、recordset 对象的使用。

(3) 能够从 ASP 界面中调取 SQL Server 数据记录。

三、实验内容及要求

(1) 建立数据库，创建数据表。

(2) 制作访问数据库界面。
(3) 进行向数据库添加数据操作。
(4) 进行修改数据库中的数据操作。
(5) 进行删除数据库中的数据操作。

四、实验方法及步骤

1. 建立数据库，创建数据表

操作步骤：打开 SQL Server，启动查询分析器，在查询分析器界面中加入代码：

```
Create database vote                          —创建投票数据库 vote
Create table voteMaster                       —创建投票项目表 voteMaster
(
   voteID int primary key,                    —编号
   voteTitle varchar(100) not null,           —投票项目
   voteSum int default 0                      —总票数
)
Insert into voteMaster values(1,'选举班长!',0) |       —向投票项目表插入数据 1
Insert into voteMaster values(2,'对网站建设的意见',0) —向投票项目表插入数据 2
Create table voteDetails                              —创建投票内容表 voteDetails
(
VoteID int foreign key references voteMaster(voteID),
                          —建立投票项目表和投票内容表的关联关系，通过 voteID 产生关联
    VoteDetailsID int not null,                       —投票内容 ID
    voteItem varchar(20) not null,                    —投票内容
    voteNum int default 0                             —投票数
primary key(voteID,voteDetailsID)                     —联合主键
)
—对"选举班长"插入记录
Insert into voteDetails values(1,1,'李明',0);         —向投票内容表插入数据 1
Insert into voteDetails values(1,2,'刘华',0);         —向投票内容表插入数据 2
Insert into voteDetails values(1,3,'郭凯',0);         —向投票内容表插入数据 3
—对"网站建设意见"插入记录
Insert into voteDetails values(2,1,'非常好',0);       —向投票内容表插入数据 4
Insert into voteDetails values(2,2,'好',0);           —向投票内容表插入数据 5
Insert into voteDetails values(2,3,'一般',0);         —向投票内容表插入数据 6
Insert into voteDetails values(2,4,'需要改进',0);     —向投票内容表插入数据 7
create trigger aa  on voteDetails  for update 0
—创建触发器使投票数据能够二表同步更新
as  begin
update voteMaster set voteSum = voteSum + 1 where voteID = (SELECT top 1 voteID from inserted)
end
```

2. 制作访问数据库的界面

设置的对象内容如下：

投票按钮：text = '投票',id = 'btnVote'

```
结果按钮: text = '结果',id = 'btnShowResult'
Radiobuttonlist: id = 'rBtnItems'
```

在 Form 表单界面中，action 转向动作为数据库连接页和数据库记录校验页面。

3. 制作连接数据库文件 connect.asp

```
<%
Dim cnobj,strcon                                          //定义变量
set cnobj = Server.CreateObject("ADODB.Connection")   //创建连接方法
strcon = "Driver = {SQL Server};Server = localhost;DataBase = vote;Uid = sa;Pwd = "
//设置服务器的登录环境,SQL Server 驱动,本地数据库,数据库名 vote,SQL Server 用户名 SA,密码为空
cnobj.Open strcon                                         //打开连接
%>
```

4. 创建数据库查询、添加、修改、删除操作(admin.asp)

1) 查询操作

```
<%
Function Check(SqlFind)         '检查对象是否存在函数
dim comm,rs
  Set comm = Server.CreateObject("ADODB.Command")      '创建 ADO DB 连接对象
    comm.ActiveConnection = cnobj
    comm.CommandType = 1
    comm.CommandText = SqlFind          '查询对象操作
    set rs = comm.Execute
    if not rs.bof and not rs.eof  then   '判断是否到达第一条记录、最后一条记录
        cnobj.close
        set cnobj = Nothing
        Check = AdminName
    else
        Check = ""
    end if
End Function
```

2) 插入操作

```
'插入数据函数
Function Admin_Insert(InsertSql)
dim comm,rs
Set comm = Server.CreateObject("ADODB.Command")
    comm.ActiveConnection = cnobj
    comm.CommandType = 1
    comm.CommandText = InsertSql     '插入对象操作
    comm.Execute
    Admin_Insert = "操作成功!"
End Function
```

3）修改操作

```
'更新信息
Function Admin_Update(UpdateSql)
dim comm,rs
Set comm = Server.CreateObject("ADODB.Command")
    comm.ActiveConnection = cnobj
    comm.CommandType = 1
    comm.CommandText = UpdateSql  '更新数据的 SQL 对象操作
    comm.Execute
    Admin_Update = "操作成功!"
End Function
```

4）删除操作

```
'信息删除
Function Admin_Delete(DeleteSql)
dim comm,rs
  Set comm = Server.CreateObject("ADODB.Command")
    comm.ActiveConnection = cnobj
    comm.CommandType = 1
    comm.CommandText = DeleteSql          '删除 SQL 对象操作
    comm.Execute
    Admin_Delete = "删除成功!"
End Function
```

五、操作技巧与注意事项

（1）从 SQL Server 中创建表的操作，从查询分析器中输入 SQL 语句操作。
（2）使用 SELECT 查询语句查询数据库中的记录。
（3）使用 UPDATE 更新语句更新数据库中的记录。
（4）使用 INSERT 插入语句插入数据库中的记录。
（5）使用 DELECT 删除语句删除数据库中的记录。

六、开发知识背景——ADO 技术

ADO 是微软强大的数据访问接口。它可为任何数据源提供高性能的访问，包括关系型数据库、非关系型数据库、电子邮件、文件系统、文本和图形以及自定义业务对象等。

用 ADO 访问数据的特点可概括如下：

ADO 是高层数据库访问技术，相对于 ODBC 来说，具有面向对象的特点。和 OLE DB 一样，可以访问多种数据源，使应用程序具有很好的通用性和灵活性，访问数据源效率高，方便 Web 应用。ADO 可以以 ActiveX 控件的形式出现，这就大大方便了 Web 应用程序的编制。

ADO 支持 Visual C++、Visual Basic、VBS、JS 等。ADO 的对象 Connection 用于表示和数据源的连接，以及处理一些命令和事务。ADO 的对象 Command 用于执行某些命令来进行诸如查询、修改数据库结构的操作。ADO 的对象 Recordset 用于处理数据源的表格集，

它是在表中修改、检索数据的最主要的方法。ADO 的对象 Field 描述数据集中的列信息。ADO 的对象 Parameter 用于对传递给数据源的命令赋参数值。ADO 的对象 Set 是一种可以方便地包含其他特殊类型对象的对象类型。ADO 提供 4 种类型的集合：

- Connection 对象具有 Error 集合。
- Command 对象具有 Parameter 集合。
- Recordset 对象具有 Fields 集合。
- Connection、Command、Recordset、Field 对象都具有 Property 集合。

常用的数据库连接方法：

1）通过 JET 数据库引擎对 Access 数据库的连接

```
m_pConnection.Open("Provider = Microsoft.Jet.OLEDB.4.0;Data Source = C:\\test.mdb","","",
adModeUnknown);
```

2）通过 DSN 数据源对任何支持 ODBC 的数据库进行连接

```
m_pConnection.Open("DataSource = adotest;UID = sa;PWD = ;","","",adModeUnknown);
  /m_pConnection - > Open("DSN = test;","","",0);
//连接叫做 test 的 ODBC 数据源
```

3）不通过 DSN 对 SQL Server 数据库进行连接

```
m_pConnection.Open("driver = {SQLServer};Server = 127.0.0.1;DATABASE = vckbase;UID = sa;PWD =
139","","",adModeUnknown);
//其中,Server 是 SQL 服务器的名称,DATABASE 是库的名称
```

七、思考题与练习

(1) 在 SQL Server 中创建数据库的操作步骤是什么?
(2) 创建 ADO 连接对象,进行查询的操作是什么?
(3) 创建 ADO 连接对象,进行修改的操作是什么?
(4) 创建 ADO 连接对象,进行删除的操作是什么?
(5) 创建 ADO 连接对象,进行添加的操作是什么?

15.10 实训 10——JSP 连接 Access 数据库

一、实验类别

操作型实验。

二、实验目的

(1) 熟悉 JSP 留言板的原理和方法。
(2) 了解 JDBC 驱动用法。
(3) 能够从 JSP 界面中调取 Access 数据记录。

三、实验内容及要求

(1) 建立 Access 数据库,创建数据表。
(2) 制作留言界面。
(3) 制作连接数据库操作。
(4) 向数据库添加留言操作。
(5) 显示留言操作。

四、实验方法及步骤

1. 建立数据库,创建数据表

操作步骤:打开 Access 软件,创建空数据库 db1,创建留言信息表,如表 15-3 所示。

表 15-3　留言信息表

字　段　名	类　　型	说　　明
id(主键)	自动编号	id 编号
name	文本	留言者姓名
theme	文本	留言主题
message	备注	留言内容

id 表示留言编号;name 表示留言者姓名;theme 表示留言主题;message 表示留言内容。操作时,需要将 id 编号设置成主键,唯一标识留言记录。

2. 制作留言界面的界面文件 lyb.jsp

设置的对象内容如下。
用户名:

```
type = "text" name = "username" value = "<% = name %>
```

留言主题:

```
type = "text" name = "theme" id = "theme" size = "46"
```

留言内容:

```
textarea name = "message" id = "message" cols = "45" rows = "5"
```

在 Form 表单界面中,用户名 value 值获取的是登录后的用户名。定义的代码如下所示:

```
String name = (String) session.getAttribute("login");。
```

确定提交:

```
name = "submit" type = "submit" class = "button" value = "确定提交"
```

在确定按钮处设置一个按钮隐藏域,代码如下所示:

```
< input name = "action" type = "hidden" value = "useredit">
```

3. 制作连接数据库文件 dellyb.jsp

JSP 连接 Access 数据库是通过调用 JDBC 连接数据库,代码如下所示:

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver").newInstance();  //新建立对象连接实例
String url = "jdbc:odbc:db1";                                 //连接数据库 db1
String user = "";                                             //设定数据库用户名和密码
String pas = "";
con = DriverManager.getConnection(url, user, pas);            //建立连接
```

4. 向数据库添加留言操作(lybadd.jsp)

```
Statement stmt = conn.createStatement();      //创建连接
String sql = "insert into lyb(name,theme,message) values('"
//向 lyb 表中插入数据记录,包括留言姓名、留言主题、留言信息
          + username                          //留言姓名
          + "','"
          + theme                             //留言主题
          + "','"
          + message                           //留言信息
           + "')";
int req = stmt.executeUpdate(sql);
```

5. 显示留言操作界面文件(showlyb.jsp)

```
int intPageSize;                              //一页显示的记录数
int intRowCount;                              //记录总数
int intPageCount;                             //总页数
int intPage;                                  //待显示页码
String strPage;
int i,j,k;
try{                                          //设置一页显示的记录数为 4 页
intPageSize = Integer.parseInt(request.getParameter("pp"));
}catch(NumberFormatException e)
{intPageSize = 5;}
strPage = request.getParameter("page");       //取得待显示页码
if(strPage == null){ //表明在 QueryString 中没有 page 这个参数,此时显示第一页数据
  intPage = 1;}
else{                                         //将字符串转换成整型
   i = 0;
   while(rs.next()){i ++ ;}
   intRowCount = i;//总行数
   intPageCount = (intRowCount + intPageSize - 1) / intPageSize;  //记算总页数
if(intPage > intPageCount) {intPage = intPageCount;}             //调整待显示的页码
<% if(intPage > 1){ %>
<a href = "showlyb.jsp?page = 1">第一页</a>
else
out.println("第一页");                        //显示“第一页”
out.println("上一页");                        //显示“上一页”
out.println("下一页");                        //显示“下一页”
out.println("最后一页");                      //显示“最后一页”
```

五、操作技巧与注意事项

(1) 掌握按钮的转向操作。

(2) 掌握向数据库添加留言的操作。

(3) JSP 连接 Access 时,采用 JDBC 的连接方法。
(4) 实现上一页和下一页的翻页功能。
(5) 在 Access 数据库中设置主键字段的操作。

六、开发知识背景——JDBC 技术

JDBC(Java data base connectivity)是一种用于执行 SQL 语句的 Java API,可以为多种关系数据库提供统一访问,它由一组用 Java 语言编写的类和接口组成。有了 JDBC,向各种关系数据库(Access、SQL Server、Oracle 等)发送 SQL 语句就是一件很容易的事。程序员只须用 JDBC API 编写一个程序就够了,它可向相应数据库发送 SQL 调用。

1. 装载驱动程序

建立 DBMS 的连接包括两个步骤:装载驱动程序和建立连接。

装载驱动程序只需要一行代码。例如,要使用 JDBC-ODBC 桥驱动程序,可以用下列代码装载:

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

驱动程序文档将告诉你应该使用的类名。例如,如果类名是 jdbc. DriverXYZ,将用以下代码装载驱动程序:

```
Class.forName("jdbc.DriverXYZ");
```

加载 Driver 类后,即可用来与数据库建立连接。

用驱动程序类与 DBMS 建立一个连接,代码如下:

```
Connection con = DriverManager.getConnection(url, "myLogin", "myPassword");
```

其中,url 是数据源名字或数据库系统;myLogin 及 myPassword 是登录 DBMS 的用户名及口令。例如,登录数据库系统的用户名为 Fernanda,口令为 J8,建立一个连接:

```
String url = "jdbc:odbc:Fred";
Connection con = DriverManager.getConnection(url,"Fernanda", "J8");
```

2. 创建对象和执行语句

Statement 对象用于把 SQL 语句发送到 DBMS。例如,利用 Connection 对象 con 创建 Statement 对象 stmt:

```
Statement stmt = con.createStatement();
```

该例中,stmt 已经存在,但还没有把 SQL 语句传递到 DBMS。需要 SQL 语句作为参数提供给 Statement 的方法。

在数据库的数据操作方法上,要想将数据传递到数据库管理系统上,操作方法是不同的,如对于 SQL 语句中的 select 语句来说,使用 executeQuery 传递到 DBMS。如果使用 create 和 update 语句,那么要使用 executeUpdate 传递到 DBMS。

3. 在表中输入数据和查询数据

使用 insert into 语句插入数据记录，例如：

```
stmt.executeUpdate("INSERT INTO COFFEES " + "VALUES ('French_Roast', 49, 8.99, 0, 0)");
```

使用 SELECT 语句获取值，例如使用 SQL 语句选择整个表数据：

```
SELECT * FROM COFFEES
```

五、思考题与练习

（1）简述 Access 中创建数据库的方法。
（2）简述留言界面文件的制作方法。
（3）简述显示留言内容文件的制作方法。
（4）理解数据库中操作数据的方法。

15.11 实训 11——ASP 在线考试系统的开发实验

一、实验类别

综合型实验。

二、实验目的

（1）掌握数据库设计的基本技术，熟悉数据库设计的每个步骤中的任务和实施方案，并加深对数据库系统概念和特点的理解。
（2）初步掌握基于 B/S 的数据库应用系统分析、设计和实现方法。
（3）初步掌握使用 Dreamweaver＋IIS 开发数据库应用系统的过程。
（4）进一步提高学生分析与解决问题的综合能力。

三、实验内容

要求学生根据周围的实际情况，深入到在线考试系统应用项目中，进行系统分析、数据库设计、系统设计以及系统实现。

四、实验要求

提交实验报告，报告由以下三部分组成：
（1）系统需求说明。简要说明系统的数据需求和功能需求。
（2）数据库结构设计。使用 E-R 图表示对数据库中要存储的信息及语义进行详细描述（即概念结构设计）；详细描述将 E-R 图转换成关系模型的过程以及对关系模式的规范化，至少达到 3NF（即逻辑结构设计）；最后以表格形式详细描述系统需要的基本表及属性、视

图和索引,对基本表的主键、候选键、外键及参照表进行说明,对基本表中数据的约束条件进行说明。

(3) 数据库的操作和应用要求。要求详细描述数据库的数据操作、处理方法和处理流程。系统应具有身份验证,对数据库数据的增加、删除、修改、查询、浏览、输出等维护功能。

(4) 系统实现。前台使用 Dreamweaver+IIS,后台使用 SQL Server 2000 数据库管理系统。

(5) 本次实验总结。

(6) 附系统执行的主要界面以及源程序。

五、实验原理

1. 系统分析与设计

(1) 数据库设计的 4 个阶段:需求分析(分析用户要求)、概念设计(信息分析和定义)、逻辑设计(设计实现)和物理设计(物理数据库设计)。

(2) 系统体系结构与功能的设计。

2. 系统实现

(1) 创建数据库及其数据表并输入一些记录。

(2) 创建工程文件及窗体,添加所需的控件,为相应控件的事件编写代码,以实现目标功能。

(3) 系统调试与测试。

(4) 经编译生成可执行文件。

六、注意事项

(1) 在 E-R 图中注意实体间联系类型的表示。

(2) 在数据库设计中应避免数据冗余(实体集的冗余和属性冗余)。

(3) 在数据库表中属性应是不可再分的最小项。

(4) 各表的联系应通过外键实现。

(5) 数据库访问接口使用 ODBC。

七、在线考试系统案例

在线考试系统具体操作步骤如下(以制作登录界面为例):

(1) 启动 Dreamweaver,并建立 ASP 文件,命名为 adminLogin.asp。如图 15-48 所示给出在线考试系统整体的 E-R 图设计。

(2) 打开 SQL Server 创建数据库,根据在线考试系统要求创建 13 张表:SuperInfo、MajorInfo、CouInfo、ClassInfo CouChoose、TeachCou、TeachInfo、StuInfo、PaperList、StuScore、PaperChoose、PaperUnCheck、PaperChecked(在 SQL Server 中创建表的方法请参考原教材第 8 章)。

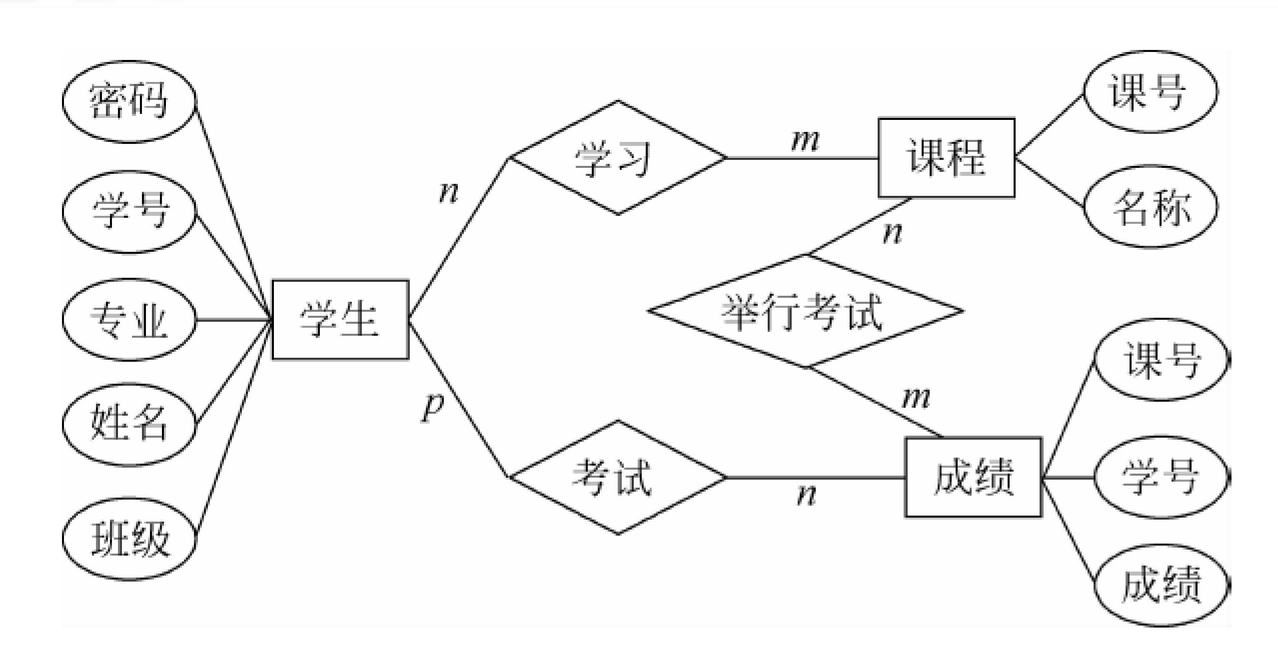


图 15-48 在线考试系统 E-R 图

① SuperInfo 表，即为管理员信息表，如表 15-4 所示。

表 15-4 SuperInfo

字段名称	内 容	类 型	长 度	备 注
SuperName	管理员姓名	Varchar	50	主键
SuperPwd	登录密码	Varchar	50	

② MajorInfo 表，即为专业表，如表 15-5 所示。

表 15-5 MajorInfo

字段名称	内 容	类 型	长 度	备 注
MajorNum	专业代号	Varchar	20	主关键字
MajorName	专业名称	Varchar	50	

MajorNum(专业号)字段为该表的主关键字，唯一地标识了一个专业的记录，与学生信息表、选课表中的 MajorNum 字段相关联。

③ CouInfo 表，即为课程信息表，如表 15-6 所示。

表 15-6 CouInfo

字段名称	内 容	类 型	长 度	备 注
CouNum	课程代号	Varchar	20	主关键字
CouName	课程名称	Varchar	50	
CouBook	学时数	Varchar	50	

CouNum 作为此表的主键，唯一标识一门课程，与选课表、授课表、选题表、成绩表中的 CouNum 关联。

④ ClassInfo 表，即为班级信息表，如表 15-7 所示。

表 15-7 ClassInfo

字段名称	内 容	类 型	长 度	备 注
ClassNum	班级代号	Varchar	12	主键
ClassName	班级名称	Varchar	50	
MajorNum	专业代号	Varchar	50	外键

MajorNum 作为本表的外键与专业表中的 MajorNum 关联(级联删除和更新)。

⑤ CouChoose 表,即为选课表,如表 15-8 所示。

表 15-8　CouChoose

字段名称	内　容	类　型	长　度	备　注
MajorNum	专业代号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键

⑥ TeachCou 表,即为授课表,如表 15-9 所示。

表 15-9　TeachCou

字段名称	内　容	类　型	长　度	备　注
TeachNum	教师工号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键
ClassNum	班级代号	Varchar	20	外键

⑦ TeachInfo 表,即为教师信息表,如表 15-10 所示。

表 15-10　TeachInfo

字段名称	内　容	类　型	长　度	备　注
TeachNum	教师工号	Varchar	20	主关键字
TeachName	教师姓名	Varchar	50	
TeachPWD	登录密码	Varchar	30	

TeachNum 作为此表的主关键字与授课表和题库表的 TeachNum 字段外键关联。

⑧ StuInfo 表,即为学生信息表,如表 15-11 所示。

表 15-11　StuInfo

字段名称	内　容	类　型	长　度	备　注
StuNum	学生学号	Varchar	20	主关键字
StuName	学生姓名	Varchar	20	
MajorNum	专业代号	Varchar	20	外键
ClassNum	班级代号	Varchar	20	外键
StuPwd	登录密码	Varchar	30	
LoginOrNot	登录状态	Varchar	10	

⑨ PaperList 表,即为题库表,如表 15-12 所示。

表 15-12　PaperList

字段名称	内　容	类　型	长　度	备　注
PaperNum	试卷代号	Varchar	100	主关键字
PaperName	试卷名称	Varchar	100	
TeachNum	教师工号	Varchar	20	

⑩ StuScore 表，即成绩表，如表 15-13 所示。

表 15-13 StuScore

字段名称	内 容	类 型	长 度	备 注
StuNum	学生学号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键
TestDate	考试时间	Varchar	11	
Score	成绩	Varchar	3	

⑪ PaperChoose 表，即选题表，如表 15-14 所示。

表 15-14 PaperChoose

字段名称	内 容	类 型	长 度	备 注
ClassNum	班级代号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键
PaperNum	试卷代号	Varchar	100	
TestDate	考试时间	Varchar	20	
StartTime	开考时间	Varchar	20	
EndTime	结束时间	Varchar	20	

⑫ PaperUnCheck 表，即未阅试题记录表，如表 15-15 所示。

表 15-15 PaperUnCheck

字段名称	内 容	类 型	长 度	备 注
StuNum	学生学号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键
TermNum	题号	Varchar	10	
StuAnswer	答案	Varchar	1000	
Score	成绩	Varchar	3	

⑬ PaperChecked 表，即已阅试题表，如表 15-16 所示。

表 15-16 PaperChecked

字段名称	内 容	类 型	长 度	备 注
StuNum	学生学号	Varchar	20	外键
CouNum	课程代号	Varchar	20	外键
TermType	试题类型	Varchar	10	
StuAnswer	答案	Varchar	1000	

(3) 在管理员登录界面 adminLogin.asp 中制作如图 15-49 所示的表格和图形。

操作方法：首先进入管理员登录界面，用户名输入 admin，密码初设为 admin，验证码输入 8842。本页面显示系统名称、管理员用户名、密码、验证码信息。尤其是验证码，采用自动随机生成技术。本系统的用户分为三类：管理员、教师、学生。每个使用者分别在不同的页面中。当管理员输入名称、密码、对应的验证码信息之后，单击“提交”按钮就会将输入信



图 15-49　登录界面

息提交给系统后台数据库，以验证用户身份。如果用户身份正确，则会进入系统的管理员控制界面。

在登录过程中，不管是一般用户还是管理员，都有各自设定的用户和密码来登录系统。如果在登录过程中，所登录的用户和密码任何一个与原始数据不一致，将不能正常地登录系统，直到所输入的数据与原始数据完全一致为止。

表 15-17 列出了管理员登录界面的表单设置情况。

根据表 15-17 所列内容，依次设置 adminLogin.asp 网页中的各个对象的值。该表单中主要涉及 3 种控件：label 标签、单行文本框、按钮控件(提交按钮和重置按钮)。设置一个 5 行 2 列的表格，然后将控件依次拖入到表格中，对表格对象设置 background 属性，将背景图片引入。通过以上设置，就能得到如图 15-49 所示的管理员登录界面。

表 15-17　管理员界面的设置

对　象　名	属　性　名	属　性　值
Label1	text	姓名
	id	空
Label2	text	密码
	id	空
Label3	text	验证码
	id	空
Text1	id	txtUserName
	text	空值
Text2	id	txtUserPassWord
	type	password
	text	空值
Text3	id	txtUserCofig
	text	空值
image	img	../zxks/back/image
	id	空值
Button1	text	提交
	id	Submit
Button2	text	重置
	id	Reset

代码 1：数字图片验证代码。

```
<td valign="bottom">
<img src="Images/checkcode.asp?t=+(new Date().getTime()" alt="点击刷新验证码"width=
"80" height="20" onClick="this.src='Images/checkcode.asp?t='+(new Date().getTime());
" style="margin:5px auto"/>
</td>
```

代码 2：用户名、密码、验证码的校验。

```
function check()
{
  var valid=true;                    //设置标记位
  var Uname=document.getElementById("AdminName").value; //定义获取用户名变量
  var Upwd=document.getElementById("AdminPwd").value; //定义获取密码变量
  var CheckCode=document.getElementById("checkcodebox").value;
//定义获取验证码变量
  if(Uname == "")                    //如果用户名为空
  {window.alert("用户名不能为空!"); //提示"用户名不能为空!"
   valid=false;
  }
  else if(Upwd == "")                //如果密码为空
  {window.alert("密码不能为空!");     //提示"密码不能为空!"
   valid=false;
  }
  else if(CheckCode=="")             //如果验证码为空
  {
   window.alert("请填写验证码!!");    //提示"请填写验证码!!"
   valid=false;
  }
  if(valid)                          //判断标记位
    return true;                     //为真,返回 true
    return false;                    //为假,返回 false
}
```

(4) 教师对成绩的管理，主要是对课程、专业、学生信息、学生成绩、试卷出题等的管理。

① 按照课程角度进行管理。

按照课程角度管理，教师可以看到课程信息。课程信息包括课程编号、课程名称、学时和课程类型。当教师双击某一课程时，可以进行添加、删除、修改等操作。

② 学生个人信息管理。

各个班级班主任老师可以进行学生个人信息管理，包括管理学号、姓名、家庭住址、联系方式等信息。

③ 学生成绩管理。

显示该课程的所有学生的成绩信息。成绩信息包括选修这门课的学生的学号、姓名、班级和考试成绩。并同时自动计算出所有学生的人数，以及所有学生的平均成绩。在成绩信息中，教师能够对学生成绩进行增加、修改和删除，并能够打印报表。

④ 试卷出题管理。

按照不同学科、不同科目的试题类型，分为以下几类：单选题、多选题、填空题、判断改

错题、问答题。由教师给出每种试题的具体分数及正确答案。同时，教师可以根据试题量的大小设定答题时间。

(5) 学生管理。

学生用户登录后，学生可以看到自己的学生信息和成绩信息。学生信息包括学号、姓名。成绩信息包括这个学生的所有课程的成绩、总分及平均分。

① 学生管理。

可以对个人信息进行维护操作。可以修改姓名、家庭住址、联系方式等信息，同时可查询个人的信息、打印个人成绩。

② 抽题操作。

学生进入系统之后，只允许根据教师设定好的课程和时间进行答题。如果提前进入，无法抽题。如果推后进入，缩短答题时间。答题结束后，学生单击“提交”按钮，系统屏幕会自动弹出本次考试得分情况。系统将考试成绩自动存入数据库中。

(6) 3 种权限的登录地址。

管理员的登录地址：http://localhost/zxks/back/adminLogin.asp。

教师的登录地址：http://localhost/zxks/back/teacherLogin.asp。

学生的登录地址：http://localhost/zxks/index.asp。

15.12　实训 12——JSP 物资管理系统的开发实验

一、实验类别

综合型实验。

二、实验目的

(1) 掌握数据库设计的基本技术，熟悉数据库设计的每个步骤中的任务和实施方案，并加深对数据库系统概念和特点的理解。

(2) 初步掌握基于 B/S 的数据库应用系统分析、设计和实现方法。

(3) 初步掌握使用 Dreamweaver+Apache Tomcat 6.0+JDK 1.6.0 开发数据库应用系统的过程。

(4) 进一步提高学生分析与解决问题的综合能力。

三、实验内容

要求学生结合物资管理公司的实际调研情况，深入到物资管理系统应用项目中，进行系统分析、数据库设计、系统设计以及系统实现。

四、实验要求

提交实验报告，报告由以下三部分组成：

(1) 系统需求说明。简要说明系统的数据需求和功能需求。

(2) 数据库结构设计。使用E-R图表示对数据库中要存储的信息及语义进行详细描述(即概念结构设计);以表格形式详细描述系统需要的基本表及属性、视图和索引,对基本表的主键、候选键、外键及参照表进行说明,对表中数据的约束条件进行说明。

(3) 数据库的操作和应用要求。要求详细描述对数据的操作、处理方法和处理流程。系统应具有身份验证,对数据库数据的增加、删除、修改、查询、浏览、输出等维护功能。

(4) 系统实现。前台使用Dreamweaver+Apache Tomcat 6.0+JDK 1.6.0,后台使用Access 2003数据库管理系统。

(5) 本次实验总结。

(6) 附系统执行的主要界面以及源程序。

五、实验原理

1. 系统分析与设计

(1) 数据库设计的4个阶段:需求分析(分析用户要求)、概念设计(信息分析和定义)、逻辑设计(设计实现)和物理设计(物理数据库设计)。需求分析采用自顶向下分析方法;概念模型设计采用自底向上的方法。

(2) 系统体系结构与功能的设计。

2. 系统实现

(1) 创建数据库及其数据表并输入记录。

(2) 创建网页文件,添加所需的控件,为相应控件的事件编写代码,实现目标功能。

(3) 系统调试与测试。

(4) 在服务器上运行实现。

六、注意事项

(1) 在E-R图中注意实体间联系类型的表示。

(2) 在数据库设计中应避免数据冗余(实体集的冗余和属性冗余)。

(3) 在数据库表中,属性应是不可再分的最小项。

(4) 各表的联系应通过外键实现。

七、物资管理系统案例

1. 系统功能分析

(1) 用户管理:主要包括用户的注册、登录、信息修改、密码修改、用户的删除、用户职务的修改。

(2) 物资入库 :主要包括物资的信息修改、添加、删除等。

(3) 物资检索:主要包括物资的查询,可通过物资的名称、价格对物资进行查询。

(4) 物资出库:主要包括物资出库的管理、出库数量、价格的计算等。

(5) 用户交流:主要包括留言板的使用和查看。

（6）物资统计：主要包括期末盘点、客户订单、系统提示。

物资管理系统功能图如图 15-50 所示。

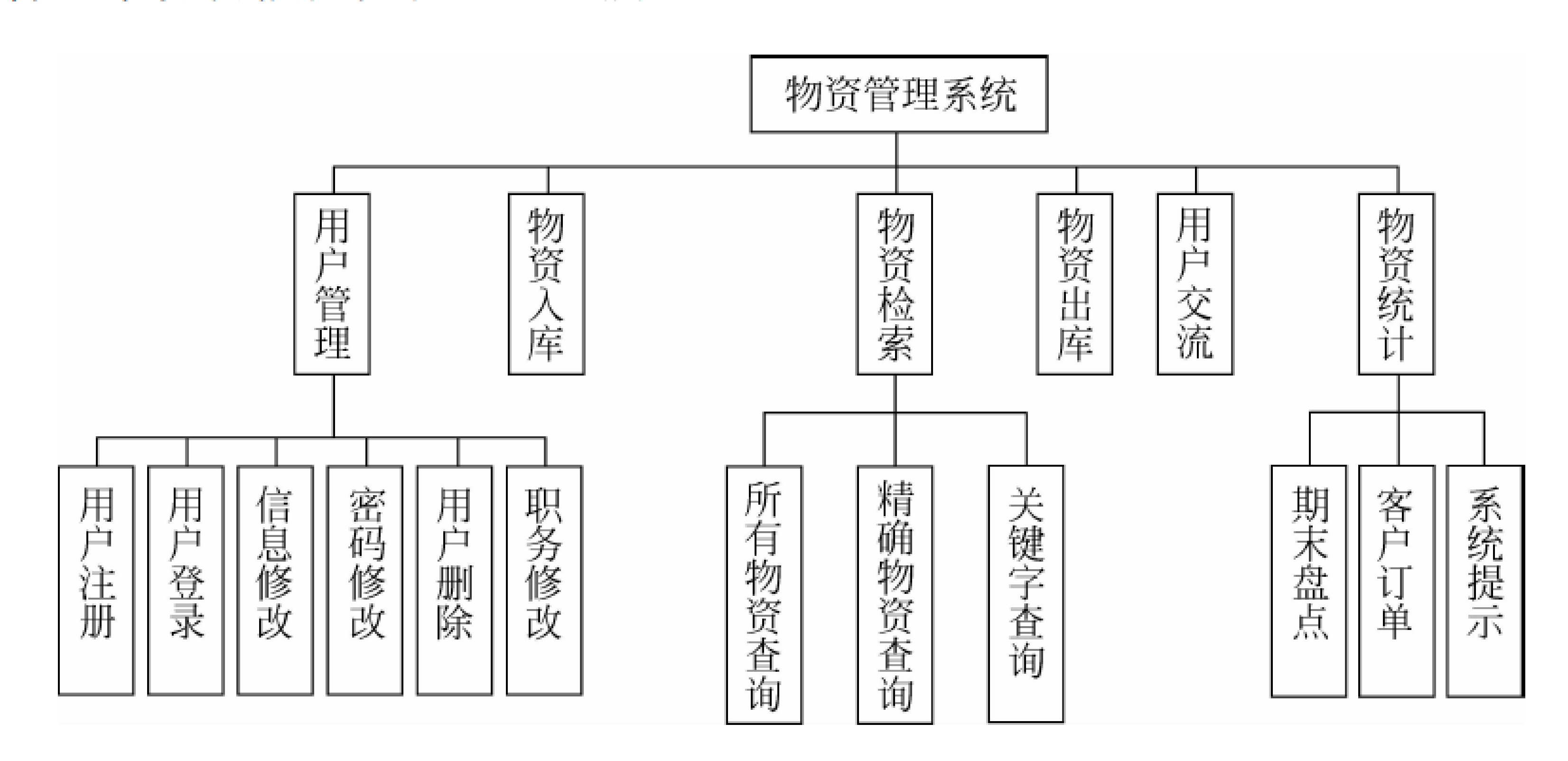


图 15-50　物资管理系统功能图

2．数据库设计

1）物资管理系统 E-R 图

物资管理系统中设计了 6 个实体，分别是物资、管理员、入库人员、出库人员、普通员工和留言板，以及入库、出库、修改、查看等联系，如图 15-51 所示。

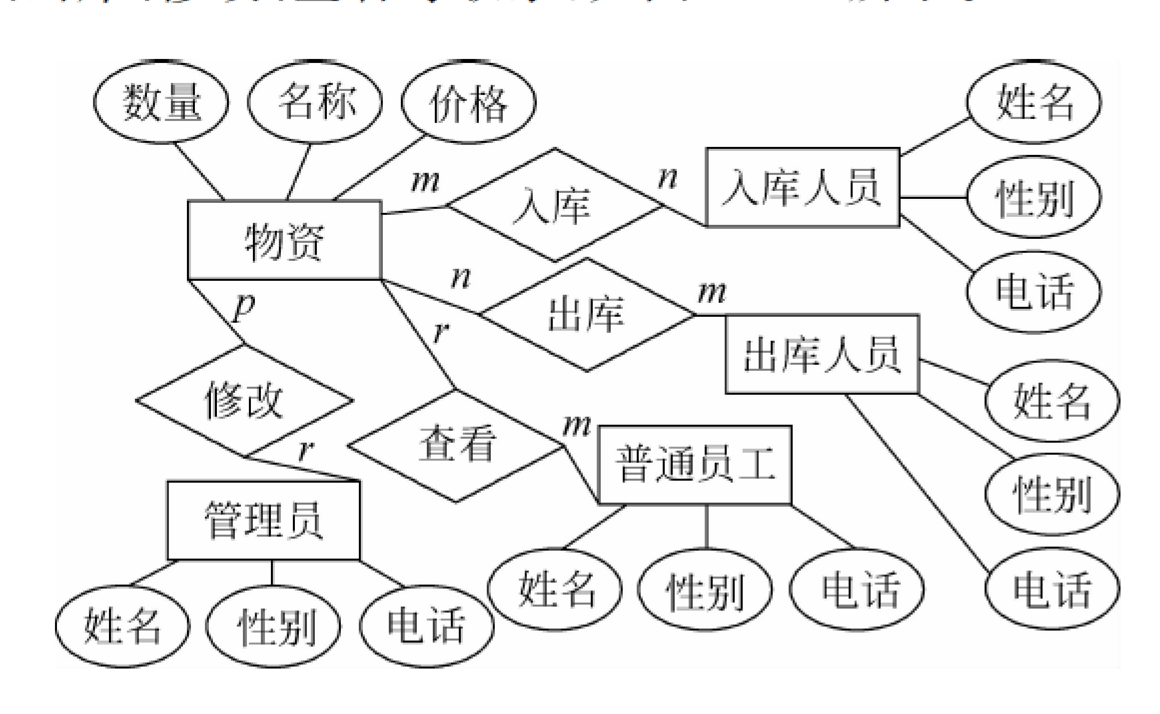


图 15-51　物资管理系统 E-R 图

2）物资管理系统数据库表设计

打开 Access 2003 数据库，创建以下表的内容：

- 物资信息表，如表 15-18 所示。

表 15-18　物资信息表

字　段　名	类　　型	说　　明
pid（主键）	数字	物资编号
pname	文本	物资名称
price	数字	物资价格
amount	数字	物资数量
date	日期/时间	入库时间

• 留言信息表，如表 15-19 所示。

表 15-19 留言信息表

字 段 名	类 型	说 明
id(主键)	自动编号	id 编号
name	文本	留言者姓名
theme	文本	留言主题
message	备注	留言内容

• 人员信息表，如表 15-20 所示。

表 15-20 人员信息表

字 段 名	类 型	说 明
id(主键)	自动编号	id 编号
name	文本	姓名
sex	文本	性别
email	文本	电子邮箱
tel	文本	电话号码
qq	文本	QQ 号码
passwd	文本	密码
birth	日期/时间	生日
zctime	文本	注册日期
zhiwu	数字	职务

• 出库物资信息表，如表 15-21 所示。

表 15-21 出库物资信息表

字 段 名	类 型	说 明
number(主键)	数字	编号
username	文本	人员姓名
pid	数字	物资编号
pname	文本	物资名称
quantity	数字	出库数量
price_new	数字	新的物资价格
cost	数字	总价格
date	日期/时间	出库日期

3. 登录界面 login. jsp

操作方法：首先进入登录界面，如图 15-52 所示，用户名输入 QQ，密码初设为 123456。本页面显示用户登录名称、用户名、密码、职务。系统的用户分为 4 类：管理员、入库人员、出库人员、普通员工。例如，管理员输入用户名、密码信息后，单击“提交”按钮就会将输入信息提交给系统后台数据库，以验证用户身份。如果用户身份正确，则会进入系统的主页界面 index. jsp。

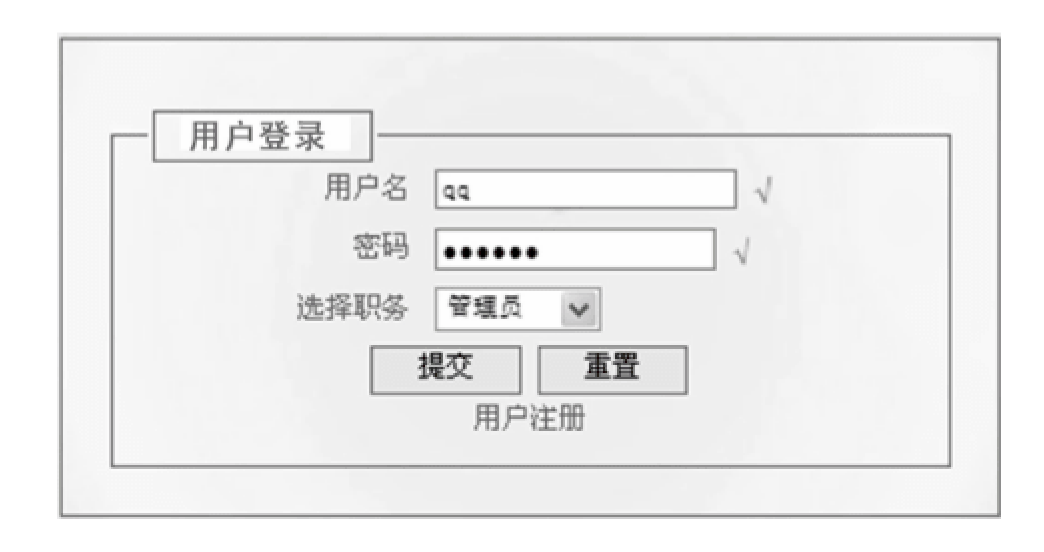


图 15-52　用户登录

登录界面的设计内容如表 15-22 所示。在登录过程中，不管是一般用户还是管理员，都使用各自设定的用户名和密码来登录系统。如果在登录过程中所使用的用户名和密码中的任何一个与原始数据不一致，将不能正常地登录系统，直到所输入的数据与原始数据完全一致为止。

表 15-22　登录界面的设计

对　象　名	属　性　名	属　性　值
Label1	text	用户名
	id	空
Label2	text	密码
	id	空
Label3	text	选择职务
	id	空
Text1	id	username
	text	空值
Text2	id	userpassword
	type	password
	text	空值
Select	id	userzhiwu
	text	管理员/入库人员/出库人员/普通员工
Button1	text	提交
	id	Submit
Button2	text	重置
	id	Reset

关键代码：

（1）对用户名、密码、职务的判断代码如下：

```
String userpasswd = request.getParameter("userpassword"); //获取密码
String userzw = request.getParameter("userzhiwu");         //获取职务
userpasswd = generatemd5string(userpasswd);                //对 userpasswd 进行 md5 加密处理
ResultSet info = stmt.executeQuery("select * from user where name = '"
          + usernames + "'");      //从数据库中查询用户名和密码是否存在
info.next();
if (!userpasswd.equals(info.getString("passwd"))) {        //判断密码是否匹配
```

```
out.println("< script language = \"javascript\"> alert(\"用户名或密码不正确...\");</script >");
  //提示用户名、密码不正确
out.println(" < script language = 'javascript'>");
out.println("location = './login.jsp'; </script >");
} else {if(! userzw.equals(info.getString("zhiwu"))) {
out.println("< script language = \"javascript\"> alert(\"用户选择职务不正确...\");
    </script >");  //提示用户选择职务不正确
out.println(" < script language = 'javascript'>");
out.println("location = './login.jsp'; </script >");}
        else if(Integer.parseInt(userzw) == 1)
        {session.setAttribute("login", usernames);
    //使用 session 对象将登录信息存入管理员表
    conn.close();   //关闭所有对象
        else if(Integer.parseInt(userzw) == 2)
        {         session.setAttribute("login", usernames);
    //使用 session 对象将登录信息存入库
        String id = info.getString("id");
```

(2) 连接数据库代码如下：

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");       //连接数据库的驱动
Connection conn = DriverManager.getConnection("jdbc:odbc:wuzi");//数据源的使用
Statement stmt = conn.createStatement();             //接收变量
info.close();                                        //关闭对象
stmt.close();                                        //关闭对象
conn.close();                                        //关闭对象
```

4. 物资入库

入库人员填写入库单，具体信息包括产品名称、产品价格、入库数量、入库日期，如图 15-53 所示。



图 15-53　物资入库单

操作方法：制作入库界面，表单中主要涉及 3 种控件，分别是 label 标签、单行文本框、按钮控件(提交按钮和重置按钮)。设置一个 5 行 2 列的表格，然后将控件依次拖入表格中，对表格对象设置 background 属性，将背景图片引入，如表 15-23 所示。

表 15-23　入库单对象设置

对　象　名	属　性　名	属　性　值
Label1	text	产品名称
	id	空
Label2	text	产品价格
	id	空
Label3	text	入库数量
	id	空
Label4	text	入库日期
	id	空
Text1	id	txtcpmc
	text	空值
Text2	id	txtcpjg
	text	空值
Text3	id	txtrksl
	text	空值
Text4	id	txtrkrq
	text	空值
Button1	text	提交
	id	Submit
Button2	text	取消
	id	Reset

代码 1：物资更新操作。

```
rs = sm.executeQuery("select * from product where pname = '" + pname + "'");
//查询 product 表中是否和新录入数据相同
    while(rs.next()){
    }String condition1 = "update product set amount = '" + amount + "',
price = '" + priceq + "'  where  pname = '" + pname + "'";
//数量更新
sm.executeUpdate(condition1);
//执行更新代码
```

代码 2：增加物资操作。

```
amount = amoun + amount;
//数量累加
String condition1 = "update product set
amount = '" + amount + "', price = '" + priceq + "'  where  pname = '" + pname + "'";
//更新 product 数据库中的数据记录
```

参 考 文 献

［1］ 张银鹤，刘治国，等. JSP 动态网站开发实践教程［M］. 北京：清华大学出版社，2007.
［2］ 周文琼，王乐球. 数据库应用与开发教程［M］. 北京：中国铁道出版社，2009.
［3］ 文东，赵俊岚，等. 数据库系统开发基础与项目实训［M］. 北京：北京科海电子出版社，2009.
［4］ 尚俊杰. 网络程序设计——ASP［M］. 北京：清华大学出版社，2009.
［5］ 蒋培，王笑梅. ASP. NET WEB 程序设计［M］. 北京：清华大学出版社，2007.
［6］ Robert Vieria. SQL Server 2008 编程入门经典［M］. 北京：清华大学出版社，2010.
［7］ 肖金秀，冯沃辉，等. JSP 程序设计教程［M］. 北京：冶金工业出版社，2003.
［8］ 潘晓南，等. 动态网页设计基础［M］. 北京：中国铁道出版社，2008.
［9］ 三味工作室. 网站设计超级 COOL——ASP［M］. 北京：人民邮电出版社，2000.
［10］ 梁建武，陈语林，等. ASP 程序设计［M］. 北京：中国水利水电出版社，2001.
［11］ 丁桂芝. ASP 动态网页设计教程［M］. 北京：中国铁道出版社，2007.

相关课程教材推荐

ISBN	书　名	定价(元)
9787302183013	IT 行业英语	32.00
9787302239659	计算机专业英语(学术能力培养)	35.00
9787302130161	大学计算机网络公共基础教程	27.50
9787302215837	计算机网络	29.00
9787302235989	数据结构(C 语言版)第 3 版　(另配套实训教材)	25.00
9787302243236	数据结构——Java 语言描述	33.00
9787302246138	计算机组成与汇编语言	29.00
9787302218555	Linux 应用与开发典型实例精讲	35.00
9787302225836	软件测试方法和技术(第二版)	39.50
9787302249177	实用软件测试教程	29.50
9787302221487	软件工程初级教程	29.00
9787302194064	ARM 嵌入式系统结构与编程	35.00
9787302202530	嵌入式系统程序设计	32.00
9787302219668	路由交换技术	29.50
9787302249559	Web 程序设计：ASP. NET	29.50
9787302227151	Web 应用程序设计实用教程	32.00
9787302237556	Java 程序设计实践教程	36.00
9787302244653	C++面向对象程序设计	35.00
9787302247487	C♯语言程序设计	23.00
9787302241171	J2EE 应用开发实例精解(WAS+RAD)	25.00
9787302228196	数据仓库与数据挖掘原理及应用	32.00
9787302245384	多媒体技术与应用	32.00
9787302241720	商务智能(第 2 版)	29.50
9787302238195	电子政务概论	36.00
9787302213567	管理信息系统	36.00

以上教材样书可以免费赠送给授课教师，如果需要，请发电子邮件与我们联系。

教学资源支持

敬爱的教师：

感谢您一直以来对清华版计算机教材的支持和爱护。为了配合本课程的教学需要，本教材配有配套的电子教案(素材)，有需求的教师可以与我们联系，我们将向使用本教材进行教学的教师免费赠送电子教案(素材)，希望有助于教学活动的开展。

相关信息请拨打电话 010-62776969 或发送电子邮件至 liangying@tup. tsinghua. edu. cn 咨询，也可以到清华大学出版社主页(http://www. tup. com. cn 或 http://www. tup. tsinghua. edu. cn)上查询和下载。

如果您在使用本教材的过程中遇到了什么问题，或者有相关教材出版计划，也请您发邮件或来信告诉我们，以便我们更好为您服务。

地址：北京市海淀区双清路学研大厦 A-707　　计算机与信息分社 梁颖　收
邮编：100084　　电子邮件：liangying@tup. tsinghua. edu. cn
电话：010-62770175-4505　　邮购电话：010-62786544